

# Eigenschaften von kritischen Booleschen Zufallsnetzwerken

## Properties of critical random Boolean networks

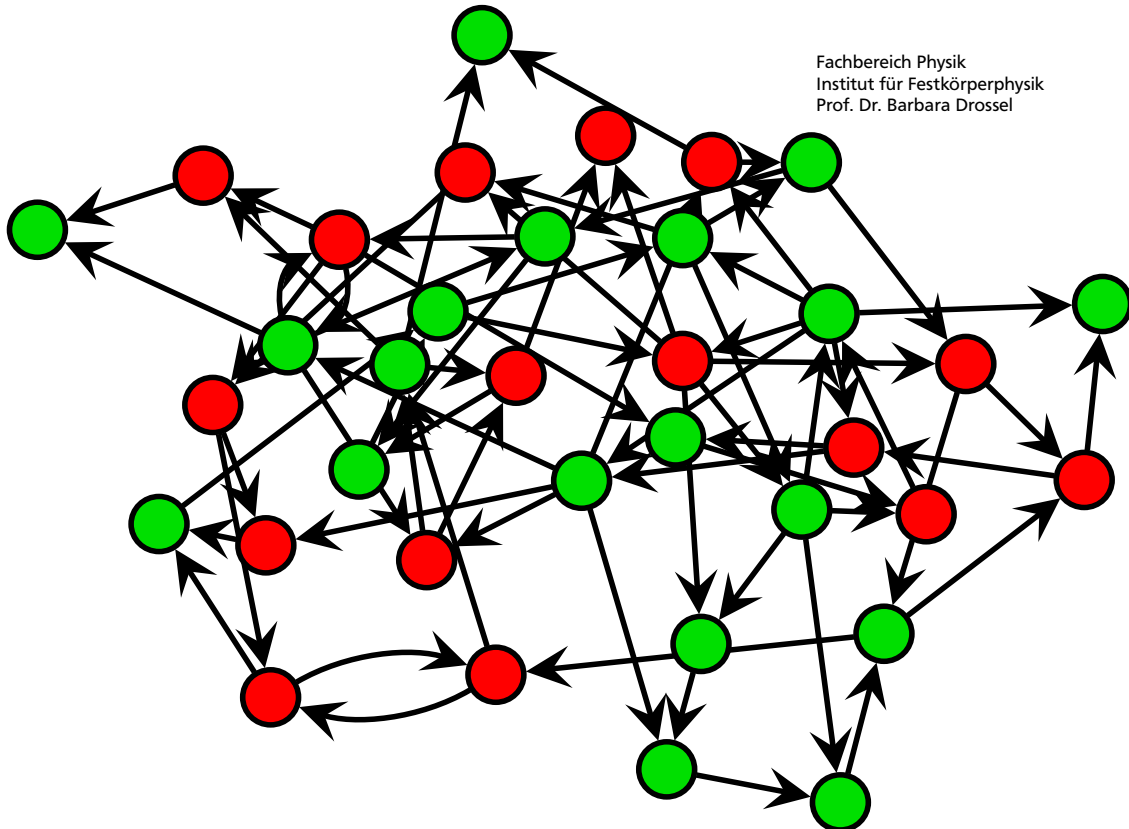
Zur Erlangung des Grades eines Doktors der Naturwissenschaften (Dr. rer. nat.)  
genehmigte Dissertation von M.Sc. Marco Möller aus Hildesheim  
Tag der Einreichung: 08.01.2015, Tag der Prüfung: 18.02.2015  
Darmstadt — D 17/2015

1. Gutachten: Prof. Dr. Barbara Drossel
2. Gutachten: Prof. Dr. Burkhard Dünweg



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Fachbereich Physik  
Institut für Festkörperphysik  
Prof. Dr. Barbara Drossel



Eigenschaften von kritischen Booleschen Zufallsnetzwerken  
Properties of critical random Boolean networks

Genehmigte Dissertation von M.Sc. Marco Möller aus Hildesheim

1. Gutachten: Prof. Dr. Barbara Drossel
2. Gutachten: Prof. Dr. Burkhard Dünweg

Tag der Einreichung: 08.01.2015

Tag der Prüfung: 18.02.2015

Darmstadt — D 17/2015

Bitte zitieren Sie dieses Dokument als:

URN: urn:nbn:de:tuda-tuprints-urn:nbn:de:tuda-tuprints-44910

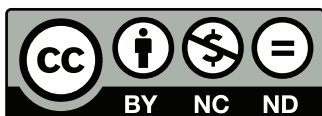
URL: <http://tuprints.ulb.tu-darmstadt.de/4491>

Dieses Dokument wird bereitgestellt von tuprints,

E-Publishing-Service der TU Darmstadt

<http://tuprints.ulb.tu-darmstadt.de>

[tuprints@ulb.tu-darmstadt.de](mailto:tuprints@ulb.tu-darmstadt.de)



Die Veröffentlichung steht unter folgender Creative Commons Lizenz:

Namensnennung – Keine kommerzielle Nutzung – Keine Bearbeitung 3.0 Deutschland

<http://creativecommons.org/licenses/by-nc-nd/3.0/de/>



---

# Erklärung zur Dissertation

Hiermit versichere ich, die vorliegende Dissertation selbstständig und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Ich versichere außerdem, dass ich bisher noch keinen Promotionsversuch unternommen habe.

Darmstadt, den 08.01.2015

---

(Möller)

---

# Kurzzusammenfassung

Boolsche Netzwerke werden häufig als generische Modelle für die Dynamik von interagierenden Einheiten genutzt, deren Zustand sich mit “an” oder “aus” gut beschreiben lässt. Neben sozialen oder ökonomischen Strukturen lässt sich beispielsweise auch die Interaktion von Genen und Proteinen durch Boolsche Netzwerke modellieren. Anhand ihrer Dynamik kann man Boolsche Netzwerke zwei verschiedenen Phasen zuordnen, der gefrorenen Phase sowie der chaotischen Phase. Im ersten Fall läuft die Dynamik des Netzwerkes unabhängig vom Anfangszustand immer auf denselben Fixpunkt. Im zweiten Fall ist sie extrem sensitiv gegenüber Störungen. Von besonderem Interesse sind die kritischen Boolschen Netzwerke, die sich genau auf der Phasengrenze befinden. Diese Arbeit beschäftigt sich mit dynamischen und strukturellen Aspekten von kritischen Boolschen Zufallsnetzen (critical random Boolean networks).

An einer Phasengrenze existiert typischerweise Skaleninvarianz und es gibt somit viele Potenzgesetze für unterschiedliche Charakteristika. Im Falle von Boolschen Netzwerken ist die Größe des gefrorenen Kerns eine solche Charakteristik. Dieser beinhaltet alle Knoten, deren Dynamik für alle Anfangsbedingungen auf demselben Fixpunkt endet. Ein bislang genutzter, sehr effizienter Algorithmus zur Bestimmung des gefrorenen Kerns geht von Knoten mit einer konstanten Aktualisierungsfunktion aus und bestimmt rekursiv weitere konstante Knoten. Es wird gezeigt, dass diese Methode scheitert, wenn der mittlere Grad eine von den verwendeten Aktualisierungsfunktionen abhängige Grenze überschreitet. Im Falle von biased Funktionen liegt diese Grenze bei 4, also durchaus im relevanten Bereich. Die Bildung des gefrorenen Kerns und dessen Robustheit wird mit Hilfe von Simulationen untersucht. Darüber hinaus wird gezeigt, dass einige wichtige Eigenschaften der Daten mittels Mean-Field-Überlegung begründet werden können.

Bei biologischen Netzwerken findet man häufig skalenfreie Ein- und/oder Ausgangsgradverteilungen. Wann immer der Exponent einer solchen skalenfreien Gradverteilung zwischen 2 und 3 liegt, divergiert das zweite Moment dieser Verteilung mit der Netzwerkgröße, was ebenfalls viele natürliche Netzwerke betrifft. Dies führt dazu, dass der Skalenexponent der Größe des gefrorenen Kerns von den Exponenten der Gradverteilungen abhängt. Des Weiteren besteht auch eine Abhängigkeit von der Art des Cut-off der Gradverteilungen. Die meisten bisherigen Ergebnisse sind für solche Netzwerke nicht gültig. Für eben diese Netzwerke werden einige Skalengesetze analytisch hergeleitet und numerisch validiert.

In verschiedenen Anwendungsgebieten existieren Netzwerke, deren Knoten sich stark durch ihre Verknüpfungen oder Dynamik unterscheiden. Dies lässt sich modellieren, indem ähnliche Knoten zu Blöcken zusammengefasst

---

werden. Hierdurch entstehen viele neue Freiheitsgrade in der Modellierung solcher Ensembles. In allgemeinen theoretischen Untersuchungen zu solchen Booleschen Blocknetzwerken stellt sich neben der Frage nach deren Kritikalität auch jene nach der sinnvollen Wahl der zahlreichen Parameter. Eine Möglichkeit, die in dieser Arbeit untersucht wird, ist die Maximierung der Entropie der Ensembles, wobei zwischen struktureller und funktionaler Entropie unterschieden werden muss. Je nach Gewichtung zwischen diesen Entropien und nach mittlerem Grad finden sich für die Struktur der gefundenen Blocknetzwerke mehrere deutlich unterscheidbare Phasen, für deren Ursprung eine Theorie entwickelt wird. Neben einer vollkommen zufälligen Phase kommen auch verschiedene geordnete vor, wie etwa eine core-periphery ähnliche Struktur und eine abgeschwächte Zwei-Gruppen-Struktur. Solche einfachen Makrostrukturen sind die wahrscheinlichsten, solange es außer Kritikalität keine Nebenbedingungen oder zusätzlichen Optimierungskriterien gibt.

---

# Abstract

Boolean networks are often used as generic models for the dynamic of interacting objects whose state can be described as “on” or “off”. Beside social or economical structures, Boolean networks can be used for example to model the interaction of genes and proteins. Based on their dynamics, Boolean networks can be assigned to different phases, either the frozen or the chaotic phase. In the first case, the dynamic of the network will always end in the same fixed point regardless its initial state. In the second case, it is very sensitive to disturbances. So-called critical networks, being at the phase boundary, are typically of special interest. This work deals with dynamical and structural aspects of critical random Boolean networks.

At a phase boundary, scale invariance exists typically and thus there are many power laws for different characteristics of the system. In case of Boolean networks, such a characteristic is the size of the frozen core. This core contains all nodes whose dynamic will end for all initial conditions on the same fixed point. A previously used, very efficient algorithm for obtaining the frozen core starts from the nodes with constant update functions and determines recursively further constant nodes. It is demonstrated that this method fails when the number of inputs per node exceeds a limit which depends on the set of update functions. In case of biased functions, this limit is 4 and therewith quite in a relevant region. The process of formation of the frozen core and its robustness was studied by computer simulation. Furthermore, it is shown that several important features of the data can be derived by a mean-field calculation.

Biological networks often show a scale free in- and / or out-degree distribution. Whenever the power-law exponent is between 2 and 3, the second moment of the distribution diverges with the network size, which also often occurs in nature. As a result, the scaling exponent of the size of the frozen core depends on the exponents of the degree distributions. Furthermore, there is a dependency on the type of cut-off of the degree distributions. Most previous results are not valid for such networks, and a set of new scaling laws is presented in this work. These scaling laws were derived analytically and elaborately validated by numerical computations.

In different areas of application, networks occur whose nodes differ strongly in link structure or dynamic. This can be modeled by pooling similar nodes into blocks. Hereby, many new degrees of freedom for modeling such ensembles arise, and therewith the question of how to choose these many different parameters in a canonical way. The presented approach for selecting parameters consists of maximizing the entropy of the ensemble, whereby it is to differentiate between structural and functional entropy. Depending on the weighting of these entropies and on the average degree, several phases

---

can be clearly distinguished, and a theory is developed. Occurring phases range from a fully random structure to several ordered ones, including a core-periphery-like structure, and an attenuated two-group structure. In the absence of competing optimization criteria and any constraints other than criticality, such simple large-scale structures are the most likely to occur.

---

# Inhaltsverzeichnis

<b>Kurzzusammenfassung</b>	<b>II</b>
<b>Abstract</b>	<b>IV</b>
<b>Inhaltsverzeichnis</b>	<b>VI</b>
<b>1 Einleitung und Grundlagen der Booleschen Netze</b>	<b>1</b>
1.1 Modell der zufälligen Booleschen Netze . . . . .	2
1.1.1 Netzwerkstruktur . . . . .	2
1.1.2 Aktualisierungsfunktion . . . . .	3
1.1.3 Dynamik . . . . .	5
1.2 Stand der Forschung . . . . .	6
1.2.1 Zeitentwicklung . . . . .	6
1.2.1.1 Phasenraum . . . . .	6
1.2.1.2 Knotentypen . . . . .	7
1.2.2 Kritikalität . . . . .	7
1.2.2.1 Skalenverhalten . . . . .	9
1.2.2.2 Biased Funktionen . . . . .	10
1.2.2.3 Konstante und invertierbare Funktionen . . .	10
1.2.3 Bestimmung des gefrorenen Kerns ausgehend von kon-	
stanten Funktionen: Containeralgorithmus . . . . .	11
1.2.3.1 Algorithmus . . . . .	11
1.2.3.2 Stochastischer Prozess . . . . .	13
<b>2 Entstehung des gefrorenen Kerns in kritischen Booleschen Netz-</b>	<b>15</b>
<b>    werken</b>	
2.1 Einleitung . . . . .	15
2.2 Modell . . . . .	16
2.2.1 Typisches Ergebnis der Containermethode . . . . .	17
2.2.2 Deterministisches Modell für die Containergrößen . . .	19
2.2.3 Beispielhafter Vergleich des Containeralgorithmus mit	
der echten Dynamik . . . . .	19
2.3 Computersimulationen des Entstehungsprozesses des gefrore-	
nen Kerns . . . . .	21
2.3.1 Einfrierdauer . . . . .	21
2.3.2 Gefrorene Knoten auf unterschiedlichen Attraktoren . .	22
2.3.3 Eigenschaften der Transiente . . . . .	24

2.4	Mean-Field-Überlegung für den Entstehungsprozess des gefrorenen Kerns . . . . .	27
2.5	Zusammenfassung . . . . .	30
<b>3</b>	<b>Skalengesetze in kritischen Booleschen Zufallsnetzen mit allgemeinen Ein- und Ausgangsgradverteilungen</b>	<b>31</b>
3.1	Einleitung . . . . .	31
3.2	Modell . . . . .	32
3.3	Verallgemeinerung der Containermethode . . . . .	34
3.4	Analytische Überlegungen . . . . .	36
3.5	Erweiterung auf andere Ensembles von Aktualisierungsfunktionen . . . . .	41
3.6	Numerische Simulationen . . . . .	44
3.7	Implementierung . . . . .	50
3.8	Zusammenfassung . . . . .	53
3.9	Ausblick . . . . .	54
<b>4</b>	<b>Blocknetzwerke</b>	<b>55</b>
4.1	Einleitung . . . . .	55
4.2	Das Modell . . . . .	56
4.2.1	Kritikalität von Blocknetzwerken . . . . .	57
4.2.2	Kritikalität bei Netzwerken mit zwei Blöcken . . . . .	59
4.3	Optimierte Ensembles . . . . .	59
4.4	Numerische Methoden . . . . .	63
4.4.1	Nebenbedingungen . . . . .	63
4.4.2	Maximierung . . . . .	65
4.4.3	Normierung . . . . .	66
4.5	Numerische Ergebnisse . . . . .	68
4.6	Ursprung der Phasenübergänge . . . . .	72
4.7	Zusammenfassung . . . . .	76
4.8	Ausblick . . . . .	77
<b>5</b>	<b>Ergebnis</b>	<b>78</b>
<b>A</b>	<b>Anhang</b>	<b>80</b>
A.1	Spektralradius von nichtnegativen reellen Matrizen . . . . .	80
A.2	Details zu optimalen Lösungen der kritischen Blocknetzwerke	83
	<b>Abbildungsverzeichnis</b>	<b>88</b>
	<b>Tabellenverzeichnis</b>	<b>90</b>
	<b>Literaturverzeichnis</b>	<b>91</b>

<hr/>	
<b>Danksagung</b>	<b>95</b>
<b>Lebenslauf</b>	<b>96</b>
<b>Publikationsliste</b>	<b>98</b>



---

# 1 Einleitung und Grundlagen der Booleschen Netze

Als generisches Modell für die Dynamik von interagierenden Einheiten werden häufig Boolesche Netzwerke eingesetzt. Diese können zum Beispiel soziale oder ökonomische Netzwerke sein, oder die Interaktion von Genen oder Proteinen beschreiben [1, 2]. Wann immer sich der Zustand einer Einheit entweder auf *an* oder *aus* ohne größere Verluste an relevanter Information reduzieren lässt, ist eine Boolesche Approximation in der Lage, viele Eigenschaften der Dynamik des echten Netzwerkes wiederzugeben [3].

Die zufälligen Booleschen Netzwerke wurden von S. Kauffman [4, 2] ursprünglich vorgeschlagen, um genregulatorische Prozesse abstrakt zu untersuchen. Hier wird der Zustand eines Knotens mit der Aktivität eines Gens identifiziert und die regulatorischen Abhängigkeiten der Gene untereinander finden sich in der Netzwerkstruktur und den Aktualisierungsfunktionen wieder. Kauffman fiel auf, dass eine gewisse Klasse von rein zufälligen Netzwerken oft erstaunlich einfache periodische Dynamiken zeigt und das, obwohl die extrem großen Anzahl von möglichen Zuständen Anderes erwarten ließ. Hieraus folgerte er, dass auch eine zumindest anfänglich zufällige Evolution zu geordneten Verhalten führt.

Eine der Eigenschaften, die mittels Boolescher Dynamik beschrieben werden können, ist der Übergang zwischen zwei unterschiedlichen dynamischen Verhalten: Einer *gefrorenen* Phase, bei der kleine Störungen des Zustandes nach einiger Zeit verschwinden, und einer *chaotischen* Phase, in der solche Störungen exponentiell schnell anwachsen, und nach kurzer Zeit das gesamte System stören [5]. Es wurde oft postuliert, dass sich viele echte Systeme, wie z. B. genregulatorische Netzwerke, auf der kritischen Grenze zwischen diesen beiden Phasen aufhalten und dadurch Eigenschaften mit diesen beiden Phasen teilen. Eine zentrale Frage war es, wie solche Systeme zur Selbstorganisation im kritischen Zustand in der Lage sind [6, 7, 8].

Um die universellen Eigenschaften solcher Modelle zu verstehen, sind Zufallsnetze sehr gut geeignet, bei denen alle Eigenschaften zufällig anhand eines Ensembles gewählt wurden. Auch wenn klar ist, dass diese nicht genau den Netzen in den Anwendungen entsprechen, konnten hieraus schon einige interessante Erkenntnisse gewonnen werden [5]. Obwohl sich die Forschung in letzter Zeit mehr zu Netzwerken mit realistischeren Eigenschaften hin verschoben hat, sind dennoch einige wichtige Fragestellungen bei den Zufallsnetzen unbeantwortet. Einige von diesen werden in dieser Arbeit behandelt.

In diesem Kapitel möchte ich einige elementare Grundlagen sowie den für mich relevanten Stand der Forschung zu Booleschen Netzen (Abschnitt 1.1)

---

und deren Kritikalität (Abschnitt 1.2.2) zusammenfassen, auf denen ich im Verlauf meiner Arbeit aufbauen werde. Vor allem die in Abschnitt 1.2.3 vorgestellte Containermethode zur Bestimmung der nicht-gefrorenen Knoten (Abschnitt 1.2.1.2) ist für einen Großteil dieser Arbeit relevant.

Im Kapitel 2 werde ich bislang unbekannte Grenzen der Containermethode bzw. neue Eigenschaften der Einfrierdynamik von Booleschen Netzwerken mit größerem mittleren Grad aufzeigen. Des Weiteren werde ich die Containermethode in Kapitel 3 auf Netzwerke mit skalenfreien Ein- und/oder Ausgangsgradverteilungen erweitern, und hier neben numerischen auch viele analytische Ergebnisse zum Skalenverhalten der nicht-gefrorenen Knoten darlegen. Im Kapitel 4 werde ich sowohl das Konzept der kritischen Booleschen Netzwerke mit dem der stochastischen Blocknetzwerke verbinden, als auch *Nullmodelle* für Netzwerke mit maximaler Entropie präsentieren und ihre Eigenschaften diskutieren. Abschließend werde ich in Kapitel 5 die wichtigsten Ergebnisse zusammenfassen.

---

## 1.1 Modell der zufälligen Booleschen Netze

---

Ein zufälliges Boolesches Netzwerk (RBN = Random Boolean Network) besteht aus drei Elementen:

- Einem gerichteten Graphen als Modell für das Netzwerk, also der Verknüpfungsstruktur der Knoten.
- Einem Modell für die Aktualisierungsfunktionen der Knoten.
- Einem Modell für die Dynamik der Knotenzustände (*an* oder *aus*).

Da diese Arbeit von **zufälligen** Booleschen Netzen handelt, müssen die Modelle für Netzwerk und Dynamik als Wahrscheinlichkeitsverteilung für verschiedene konkrete Realisierungen gegeben sein (Ensembles). Oft sind diese Verteilungen parametrisiert, was gute Möglichkeiten der Analyse der Modelle bietet. Im Folgenden werden ich zunächst Modelle für ein einzelnes RBN beschreiben und anschließend ins Ensemble-Bild übergehen. Ein Ensemble beschreibt ganze Klassen von ähnlichen RBNs.

---

### 1.1.1 Netzwerkstruktur

---

Ein gerichteter Graph  $G = (\mathcal{N}, \mathcal{E})$  wird durch einen Knotenvektor  $\mathcal{N}$  sowie eine Multimenge von Kanten  $\mathcal{E}$  charakterisiert. Eine Kante  $(a, b) \in \mathcal{E}$  ist ein 2-Tupel von Knoten  $a, b \in \mathcal{N}$ . Die wichtigsten Kenngrößen für einen Graphen sind in dieser Arbeit dessen Anzahl an Knoten  $N = |\mathcal{N}|$  und Kanten  $E = |\mathcal{E}|$  sowie der mittlere Knotengrad  $K = \frac{E}{N}$ . Eine andere häufig benutzte Charakterisierung von Graphen stellen Adjazenzmatrizen dar. Eine Adjazenzmatrix  $A = [a_{s,r}] \in \mathbb{N}^{N \times N}$  gibt in ihren Einträgen  $a_{s,r}$  die Anzahl der Kanten vom Knoten  $s$  zum Knoten  $r$  an<sup>1</sup>. Da ich gerichtete Graphen betrachte, gibt

---

<sup>1</sup> Blocknetzwerke die ich später einführen werde haben eine entgegengesetzte Reihenfolge ihrer Indices.

es a priori keinen Zusammenhang zwischen den Einträgen  $a_{s,r}$  und  $a_{r,s}$ . Im Umfeld der RBN werden zum Teil gesonderte Anforderungen an den Graphen gestellt wie das Verbot von Schleifen ( $a_{s,s} = 0$ ) bzw. Mehrfachkanten ( $a_{s,r} \leq 1$ ). Diese Annahmen werde ich in dieser Arbeit **nicht** treffen, um die analytischen Rechnungen zu vereinfachen. Speziell im von mir betrachteten Fall  $K \ll N \Leftrightarrow E \ll N^2$  sind Schleifen und Mehrfachkanten ohnehin sehr selten<sup>2</sup>. Des Weiteren schließt mein Modell nicht zusammenhängende (reduzible) Graphen ebenfalls mit ein.

Der Grad eines Knotens bezeichnet die Anzahl der eingehenden  $k_{\text{ein}}$  bzw. abgehenden  $k_{\text{aus}}$  Kanten eines Knoten. Im Mittel über alle Knoten eines Netzwerkes sind diese Grade identisch  $K_{\text{ein}} = K_{\text{aus}} = K$ , da jede eingehende Kante eines Knotens zugleich abgehenden Kante eines anderen Knotens ist<sup>3</sup>. Speziell für größere Netzwerke lohnt es sich, auch die Wahrscheinlichkeitsverteilungen der Ein- und Ausgangsgrade  $P_{\text{ein}}(k)$  bzw.  $P_{\text{aus}}(k)$  zu betrachten. Falls ein Graph nur durch  $N$  und  $K$  charakterisiert ist und die Verknüpfungsstruktur rein zufällig (gleichverteilt) gewählt wird, ergibt sich eine Binomialverteilung für den Eingangs- bzw. Ausgangsgrad bzw. im thermodynamischen Limes  $N \rightarrow \infty$  die Poissonverteilung als

$$P_{\text{ein}}(k) = \frac{K_{\text{ein}}^k}{k!} e^{-K_{\text{ein}}} \quad (1.1)$$

$$P_{\text{aus}}(k) = \frac{K_{\text{aus}}^k}{k!} e^{-K_{\text{aus}}} . \quad (1.2)$$

In bestehenden Arbeiten zu RBNs wurde zudem oft angenommen, dass der Eingangsgrad von allen Knoten identisch  $P_{\text{ein}}(k) = \delta_{K_{\text{ein}},k}$ , und der Ausgangsgrad poissonverteilt ist.

Weite Teile dieser Arbeit werden sich mit alternativen Klassen von Netzwerken befassen. Im Kapitel 3 werde ich Netzwerke mit skalenfreien Gradverteilungen untersuchen und im Kapitel 4 wird es um Möglichkeiten gehen, ein zufälliges Netzwerk mit vorgegebenen Makrostrukturen zu versehen.

---

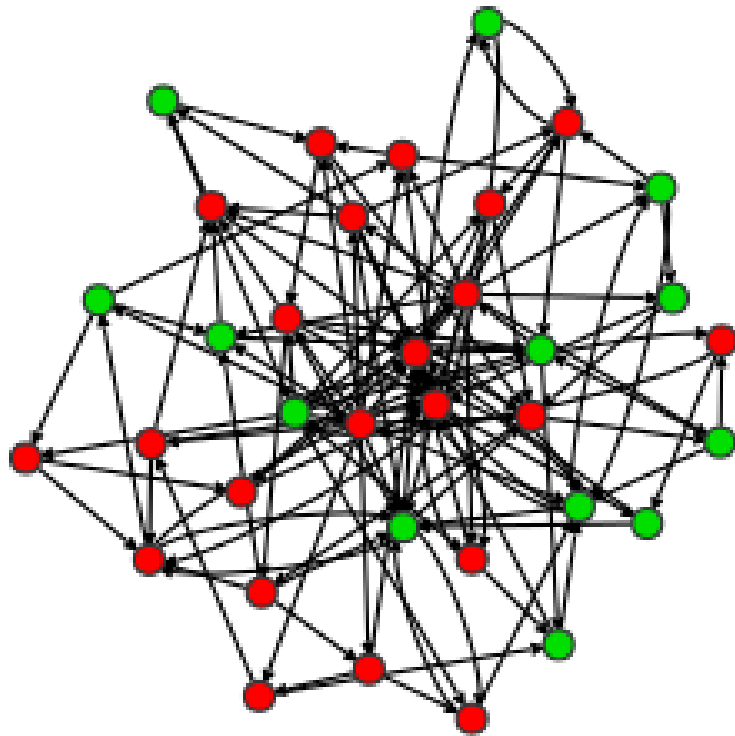
### 1.1.2 Aktualisierungsfunktion

---

Ein Boolesches Netzwerk ist über die Netzwerkstruktur hinaus noch durch zwei weitere Eigenschaften charakterisiert. Zum Einen besitzt jeder Knoten  $i$  einen Booleschen Zustand  $\sigma_i(t) \in \mathbb{F} = \{0, 1\}$ , der sich über die Zeit  $t$  ändern kann. Siehe hierzu auch Abbildung 1.1. Diese Zeit könnte kontinuierlich oder diskret gewählt sein, wobei bei diskreten Zuständen auf eine kontinuierliche Zeit verzichtet werden kann. Im Folgenden ist daher  $t \in \mathbb{N}$ . Das gesamte

<sup>2</sup> Die Adjazenzmatrix  $a_{s,r}$  eines Netzwerkes mit  $N$  Knoten hat  $N^2$  Einträge. Solange deutlich weniger Kanten als Einträge in dieser Matrix existieren, d.h.  $E = \sum_{s,r=1}^N a_{s,r} \ll N^2$ , ist die Wahrscheinlichkeit für einen Eintrag  $a_{s,r} > 0$  gering (z. B. Schleifen), noch geringer ist zudem die Wahrscheinlichkeit für Mehrfachkanten  $a_{s,r} > 1$ .

<sup>3</sup> Ich werde im Verlauf dieser Arbeit den Großbuchstaben  $K$  für mittlere Gerade verwenden, und den Kleinbuchstaben  $k$  für Grade von individuellen Knoten.



**Abbildung 1.1.: Beispiel für ein Boolesches Netzwerk**

Netzwerk mit  $N = 35$  Knoten und einem mittleren Grad  $k = 2$ . Die Farben rot/grün stehen für den aktuellen Zustand der Knoten 0/1. Der hier dargestellte Zustand ist ein zufälliger Startzustand, der Zustand nach  $t = 25$  Zeitschritten ist in Abbildung 1.2 gezeigt.

Netzwerk kann  $2^N$  unterschiedliche Zustände annehmen, der Phasenraum ist also diskret und endlich.

Zum Anderen wird jedem Knoten  $i$  eine Aktualisierungsfunktion  $f_i$  zugeordnet. Diese werden bei Zufallsnetzen analog zur Netzwerkstruktur anhand einer Wahrscheinlichkeitsverteilung zufällig gewählt. Für einen Knoten mit Eingangsgrad  $k_{\text{ein}}$  ist dies eine Abbildung  $f_i : \mathbb{F}^{k_{\text{ein}}} \rightarrow \mathbb{F}$  der Zustände der Knoten am anderen Ende der Eingangskanten  $\{\sigma_j(t)\}$  zum Zeitpunkt  $t$  auf den neuen Zustand des Knotens  $\sigma_i(t+1) = f_i(\{\sigma_j(t)\})$ . Insgesamt gibt es  $2^{2^{k_{\text{ein}}}}$  solche Funktionen. Eine typische Darstellung ist eine Wertetabelle, die für unterschiedliche Kombinationen von Eingangswerten einen Ausgangswert angibt. Solche Funktionen lassen sich in unterschiedliche Klassen unterscheiden. Eine typische Unterteilung ist:

konstant <sup>4</sup> Der Ausgang ist unabhängig von den Eingangswerten, also immer 0 oder immer 1.

invertierbar <sup>5</sup> Jede Änderung von genau einem Eingangswert hat eine Änderung des Ausgangs zur Folge. Das entspricht der XOR-Funktion oder ihrer Negation.

<sup>4</sup> auch gefroren genannt

<sup>5</sup> auch reversibel genannt

Ein		mögliche Ausgänge															
a	b	konst.		kanalisierend 1				kanalisierend 2						inv.			
0	0	0	1	0	1	0	1	1	0	0	0	0	1	1	1	1	0
0	1	0	1	0	1	1	0	0	1	0	0	1	0	1	1	0	1
1	0	0	1	1	0	0	1	0	0	1	0	1	1	0	1	0	1
1	1	0	1	1	0	1	0	0	0	0	1	1	1	1	0	1	0

**Tabelle 1.1.: Liste aller  $2^{2^2} = 16$  möglichen Aktualisierungsfunktionen für einen Knoten mit 2 Eingängen und ihre Klassifizierung**

**kanalisierend** Es existiert mindestens ein konkreter Wert eines konkreten Eingangs, so dass der Ausgang konstant ist, also nicht mehr von den verbleibenden Eingängen abhängt. Dies lässt sich für den Fall von zwei Eingängen in zwei Typen unterteilen:

1. Der Ausgang ist von mindestens einem Eingang unabhängig.
2. Der Ausgang ist nur für genau eine Eingangskombination 1 bzw. 0, ansonsten 0 bzw. 1.

In Tabelle 1.1 sind alle möglichen Aktualisierungsfunktionen für einen Knoten mit zwei Eingängen zusammen mit ihrem Typ aufgelistet.

Für das Ensemble der Aktualisierungsfunktionen benötige ich für jede mögliche Funktion einen Wahrscheinlichkeitswert. Die einfachste Annahme ist die Gleichverteilung, also dass alle  $K_{\text{ein}}$ -stelligen Funktionen jede der  $2^{2^{K_{\text{ein}}}}$  möglichen Funktionen mit der Wahrscheinlichkeit  $2^{-1 \cdot 2^{K_{\text{ein}}}}$  annehmen. Alternative typische Ensemble sind jene, die nur eine Teilmenge der möglichen Funktionen enthalten. Im Laufe der Arbeit werde ich das Ensemble der *konstanten und invertierbaren* Funktionen nutzen.

Ein anderes in dieser Arbeit genutztes Funktionenensemble sind die *biased* Funktionen. In diesem Ensemble sind alle möglichen Funktionen vertreten, allerdings nicht gleich gewichtet. Die Wahrscheinlichkeit für eine Funktion hängt von der Anzahl  $m$  der 1-en in der Ausgangsspalte ihrer Wahrheitstabelle und einem Parameter  $p \in [0, 1]$  ab. Sie berechnet sich als  $p^m (1 - p)^{K_{\text{ein}} - m}$ , ordnet also jeder 1 eine Wahrscheinlichkeit  $p$  und jeder 0 eine Wahrscheinlichkeit  $1 - p$  zu. In den Grenzfällen  $p = 0$  bzw.  $p = 1$  enthält das Ensemble ausschließlich die konstante 0 bzw. 1 Funktion. Für einen Wert von  $p = \frac{1}{2}$  sind alle möglichen Funktionen gleich häufig vertreten.

### 1.1.3 Dynamik

Es gibt zwei typische Regeln, wie die Aktualisierungsfunktionen auf den Graphen  $G$  und seinen Zustand  $\sigma$  angewandt werden können. Bei *zufälliger Aktualisierung* wird in jedem Zeitschritt  $t$  ein zufälliger Knoten  $i$  gewählt und nur dessen Zustand  $\sigma_i$  unter Anwendung seiner Aktualisierungsfunktion im Zeitschritt  $t + 1$  aktualisiert.

---

Im Gegensatz dazu wird bei der *parallelen (deterministischen) Aktualisierung* **jeder** Knoten in jedem Zeitschritt mittels seiner Aktualisierungsfunktion aktualisiert.

In einigen Arbeiten [9, 10] wird in diesem Schritt zusätzlich ein zufälliges Rauschen auf die Knotenzustände angewendet.

In dieser Arbeit wird parallele Aktualisierung ohne Rauschen verwendet.

---

## 1.2 Stand der Forschung

---

Im Folgenden möchte ich kurz einige relevante Forschungsergebnisse vorstellen, die für das Verständnis dieser Arbeit wichtig sind. Ich gehe in Abschnitt 1.2.1 im Besonderen auf die Dynamik des Netzwerks ein, sowie auf die verschiedenen dynamischen Rollen der Knoten. Im Abschnitt 1.2.2 werden unterschiedliche dynamische Phasen des Netzwerks diskutiert, und insbesondere die kritische Grenze zwischen diesen. Abschließend werde ich in Abschnitt 1.2.3 die Containermethode vorstellen, mit der sich unter Anderem der gefrorene Kern eines Netzwerkes sowohl numerisch als auch analytisch bestimmen lässt.

---

### 1.2.1 Zeitentwicklung

---

Durch das Hinzufügen einer Dynamik auf dem Zustandsraum der Booleschen Netzwerke wird sich der Zustand des Netzes im Allgemeinen über die Zeit ändern. In diesem Abschnitt werden wesentliche Eigenschaften und Merkmale vorgestellt.

---

#### 1.2.1.1 Phasenraum

---

Die Zeitentwicklung des Zustandes  $\sigma$  eines RBNs hängt maßgeblich von dem Startzustand  $\sigma(0)$  ab. In einem Netzwerk mit  $N$  Knoten enthält der Phasenraum  $2^N$  Zustände. Bei einem solchen diskreten endlichen Zustandsraum ist klar, dass sich bei paralleler (deterministischer) Aktualisierung spätestens nach  $2^N$  Zeitschritten ein Zustand wiederholt. Eine Folge sich wiederholender Zustände wird als Attraktor bezeichnet. Die minimale Anzahl von Zuständen bis zur nächsten Wiederholung wird als Länge des Attraktors bezeichnet. Ein Attraktor der Länge 1 wird Fixpunkt genannt. Die Zustände des Systems, während es den Attraktor durchläuft, werden *rekurrent* genannt. Im Gegensatz dazu werden die Zustände einschließlich des Startzustandes bis zum erstmaligen Erreichen des Attraktors als *transient* bezeichnet und ihre gesamte Folge als *Transiente*. Die Menge der Zustände, aus denen das System auf einen gewissen Attraktor läuft, werden als *Einzugsgebiet* dieses Attraktors bezeichnet. Es ist möglich, dass ein System für einen gewissen Startzustand keine bzw. eine leere Transiente besitzt, falls direkt auf einem Attraktor gestartet wird. In der Literatur werden Attraktoren, deren Einzugsgebiet nur aus dem Attraktor selbst besteht, auch teilweise als Zyklus bezeichnet, da sie nicht anziehen. Ich verzichte hier auf diese Unterscheidung.

---

Ein Netzwerk besitzt typischerweise sehr viele Attraktoren. Es konnte für kritische Netze (siehe Abschnitt 1.2.2) gezeigt werden, dass die mittlere Anzahl und mittlere Länge der Attraktoren stärker als jedes Potenzgesetz mit  $N$  anwächst [11, 12].

---

#### 1.2.1.2 Knotentypen

---

Um ein tieferes Verständnis des dynamischen Verhaltens zu erlangen, hat es sich als nützlich herausgestellt, die Knoten des Netzwerkes anhand ihres Verhaltens auf einem Attraktor zu klassifizieren. Dazu bezeichnen wir einen Knoten als *gefroren*, wenn sein Zustand in der weiteren Dynamik konstant bleibt und nennen das Erreichen dieses Zustands *einfrieren*<sup>6</sup>.

Als Erstes gibt es Knoten, die auf jedem Attraktor auf den gleichen Wert einfrieren. Diese Knoten liefern einen konstanten Eingabewert für andere Knoten und sind davon abgesehen irrelevant für die Dynamik des Netzwerkes. Sie bilden den *gefrorenen Kern*<sup>7</sup> des Netzwerkes [13].

Als zweites gibt es die etwas allgemeinere Klasse der *irrelevanten Knoten*, zu denen neben den gefrorenen Knoten noch weitere gehören. Diese zeichnen sich dadurch aus, dass ihre Ausgänge nur zu anderen irrelevanten Knoten gehen. Sie ändern zwar ggf. ihre Zustände auf einem Attraktor, beeinflussen damit aber nicht die weitere Zeitentwicklung des Attraktors.

Als drittes gibt es die *relevanten Knoten*. Diese haben auf mindestens einem Attraktor einen nicht konstanten Zustand und ihr Ausgang ist mit einem anderen relevanten Knoten verbunden. Diese relevanten Knoten bestimmen komplett allein die Anzahl und Perioden der Attraktoren und bilden den *computational-core* des Netzwerkes (siehe hierzu auch Abbildung 1.2). Die Entdeckung, dass nur die relevanten Knoten für das asymptotische Verhalten verantwortlich sind, war ein wichtiger Schritt im Verständnis von RBNs [14, 15, 16, 17, 12, 18].

---

#### 1.2.2 Kritikalität

---

Als nächstes betrachte ich, wie sich ein System mit einem leicht gestörten Anfangszustand  $\sigma^a(0)$  relativ zu seiner ungestörten Variante  $\sigma^b(0)$  mit der Zeit entwickelt. Hierfür nutze ich die Hamming-Distanz

$$h(\sigma^a, \sigma^b) = \frac{1}{N} \sum_i (\sigma_i^a - \sigma_i^b)^2 \quad (1.3)$$

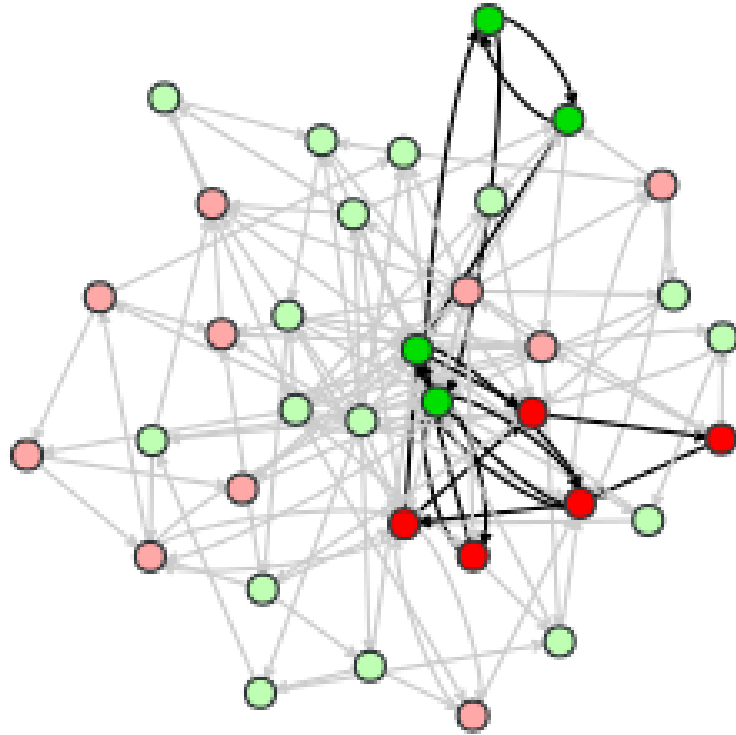
zwischen zwei Zuständen  $\sigma^a$  und  $\sigma^b$ , normiert bzgl. der Anzahl der Knoten  $N$ . Diese Distanz zählt die Stellen, an denen sich die Zustände unterscheiden. Anstelle einer direkten Untersuchung der mikroskopischen Dynamik

---

<sup>6</sup> Ein Knoten kann auch bereits auf einer Transiente einfrieren.

<sup>7</sup> Es gibt gefrorene Knoten, die nicht Teil des gefrorenen Kerns sind, aber durchaus auf manchen oder allen Attraktoren einfrieren.





**Abbildung 1.2.: Beispiel für ein Boolesches Netzwerk auf einem Attraktor**

Netzwerk mit  $N = 35$  Knoten und einem mittleren Grad  $k = 2$ . Die Farben rot / grün stehen für den aktuellen Zustand der Knoten 0/1. Dieses Netzwerk ist identisch zu dem aus Abbildung 1.1, nur dass hier der Zustand nach  $t = 25$  Zeitschritten gezeigt ist. Knoten mit blassen Farben gehören zum gefrorenen Kern und haben den hier gezeigten Wert auf jedem Attraktor. Die Knoten mit kräftigen Farben sind die irrelevanten, aber nicht-gefrorenen Knoten sowie die relevanten Knoten. Diese haben nicht auf jedem Attraktor den gleichen Wert.

von Gleichung (1.3) kann ich mit Hilfe der annealed Approximation [19] annehmen, dass die Kanten des Netzwerkes in jedem Zeitschritt anhand des Ensembles neu gewählt werden. Hiermit werden Korrelationen zwischen Knotenzuständen aufgehoben. Wenn ich mich zudem auf frühe Zeitpunkte nach der Störung beschränke, so dass  $h(t) := h(\sigma^a(t), \sigma^b(t)) \ll 1$  gilt, kann ich die Wahrscheinlichkeit vernachlässigen, dass zwei Eingänge eines Knoten gleichzeitig ihren Wert ändern (im thermodynamischen Limes für  $N^2 \gg E$ ), da diese nur die Ordnung  $\mathcal{O}(h(t)^2)$  hätte. Hiermit lässt sich die Zeitentwicklung  $h$  reduzieren auf

$$h(t + 1) \approx \lambda h(t). \quad (1.4)$$

Der Faktor  $\lambda$  wird als *Sensitivität* bezeichnet und ist eigentlich abhängig vom konkreten ungestörten Anfangszustand und dem konkreten RBN. Im Ensemble-Bild wird dieser ebenfalls mittels Erwartungswerten als mittlere Sensitivität gegeben.

Anhand des Sensitivität lässt sich ein Netzwerk oder ein ganzes Ensemble in drei Klassen unterteilen:



---

$\lambda > 1$  chaotisch Hier streben die Zeitentwicklungen von den beiden Anfangszuständen anfangs exponentiell auseinander (da sich eine Störung im Mittel auf mehr als einen Knoten je Zeitschritt ausbreitet[20]) bis sie sich schließlich auf einen maximalen Abstand einpendeln. Chaotische Netzwerke zeichnen sich typischerweise durch sehr viele Zustandsänderungen je Zeitschritt aus. Selbst im thermodynamischen Grenzfall  $N \rightarrow \infty$  bleibt ein nicht verschwindender Anteil der Netzwerke nicht-gefroren [12].

$\lambda < 1$  gefroren Hier wird die Störung über die Zeit gedämpft, sie friert also mit der Zeit ein. Gefrorene Netzwerke haben Dynamiken, die übermäßig häufig bzw. schnell auf kurze Attraktoren bzw. Fixpunkte laufen. Solche Netzwerke haben selbst im thermodynamischen Grenzfall nur eine endliche Zahl nicht-gefrorener Knoten [17], somit geht der nicht-gefrorene Anteil gegen 0.

$\lambda = 1$  kritisch Im Mittel wird durch die Zustandsänderung eines Knotens ein weiterer Knoten seinen Zustand ändern. Die Größe einer Störung (Lawine) wächst, solange sie noch nicht abklingt, typischerweise wie ein Potenzgesetz in der Zeit an.

Diese Arbeit befasst sich ausschließlich mit kritischen Booleschen Netzwerken, da diese den interessanten Grenzfall zwischen den zwei eher simplen Dynamiken darstellen [19, 21]. Wie in [22] beschrieben haben die Attraktoren solcher Netzwerke eine hohe Robustheit gegenüber äußeren Störungen, und sind trotzdem in der Lage sich unter Evolution an veränderte Rahmenbedingungen anzupassen. Zudem haben die Attraktoren eine für biologische Systeme realistische mittlere Länge, die weit unterhalb der Größe des Phasenraums  $2^N$  liegt. Zu Anfang ist in [4] aufgrund Beobachtungen in Simulationen kleiner Netze die These aufgestellt worden, dass die mittlere Anzahl und mittlere Länge der Attraktoren mit  $\sqrt{N}$  anwächst. Mittlerweile ist jedoch bekannt, dass diese stärker als jedes Potenzgesetz mit  $N$  anwachsen [23, 11, 12].

---

### 1.2.2.1 Skalenverhalten

---

Durch [14, 15, 16, 17, 12, 18] ist Einiges über das Skalenverhalten von den verschiedenen Knotentypen in kritischen Netzen bekannt. Die mittlere Anzahl an nicht-gefrorenen Knoten skaliert wie  $N_{\text{nf}} \sim N^{\frac{2}{3}}$ , und die mittlere Anzahl an relevanten Knoten wie  $N_{\text{rel}} \sim N^{\frac{1}{3}}$  falls  $K > 1$ . Der gefrorene Kern enthält alle Knoten bis auf Knoten in der Größenordnung von  $N^{\frac{2}{3}}$ .

Es wurde bereits vor Beginn dieser Promotion gezeigt, dass diese Skalengesetze nur unter gewissen Voraussetzungen gelten [24, 25]. Im Kapitel 3 werde ich auf Netzwerke eingehen, bei denen diese Gesetzmäßigkeiten so nicht mehr gelten.

---

### 1.2.2.2 Biased Funktionen

---

Um bei beliebigem gegebenen mittleren Grad  $K$  kritische Netzwerke zu erhalten, werde ich in dieser Arbeit die Funktionsmenge in Abhängigkeit von  $K$  entsprechend modifizieren. Bei den biased Funktionen muss der Wert von  $p$  nach [20] gewählt werden als

$$p = \frac{1}{2} \pm \sqrt{\frac{1}{4} - \frac{1}{2K}}, \quad (1.5)$$

um ein kritisches Netzwerk zu erhalten. Im Folgenden werde ich nur das Minuszeichen verwenden. Dies bedeutet, dass meine Aktualisierungsfunktionen durch Nullen dominiert werden und das Minderheitsbit 1 mit Wahrscheinlichkeit  $p$  angenommen wird. Diese Gleichung ist nur für  $K \geq 2$  reell, was somit die untere Schranke für kritische Netzwerke mit biased Funktionen ist. Zudem hat sie eine implizite Darstellung  $1 = K \cdot 2p(1 - p) = \lambda$ , was genau der Sensitivität entspricht. Anschaulich entspricht diese Sensitivität der Wahrscheinlichkeit, dass bei einer Zustandsänderung eines Knotens genau ein<sup>8</sup> anderer Knoten daraufhin ebenfalls seinen Zustand ändert. Eine solche initiale Änderung bei einem mittleren Grad von  $K$  breitet sich auf genau  $K$  andere Knoten aus, was dem ersten Faktor entspricht. Der Wert  $2p(1 - p)$  entspricht der Wahrscheinlichkeit, dass eine biased Funktion einen anderen Ausgangswert annimmt, wenn sich genau ein Eingangswert ändert.

---

### 1.2.2.3 Konstante und invertierbare Funktionen

---

Eine alternative Funktionenmenge, die ich verwenden werde, besteht ausschließlich aus konstanten (konstant 0 oder konstant 1) und invertierbaren Funktionen (XOR oder not XOR). Durch passende Wahl des Anteils  $r_c$  der invertierbaren Funktionen [24]

$$r_c = \frac{1}{K} \quad (1.6)$$

wird ein solches Netzwerk ebenfalls kritisch. Bei einem solchen Netzwerk können alle Knoten mit konstanter Funktion eliminiert werden, ebenso wie die Kanten von und zu ihnen, da diese die Dynamik (und damit die Kritikalität) nicht beeinflussen. Die Anzahl der übrigen Kanten ist identisch mit der Anzahl der übrigen Knoten; das Netzwerk hat also einen effektiven Grad von 1 [14, 11]. Der effektive Grad entspricht dem mittleren Grad, den ein Netzwerk besitzt, nachdem alle nicht für die Dynamik relevanten Knoten und Kanten entfernt wurden, wie beispielsweise Knoten mit konstanter Aktualisierungsfunktion. Wenn nun ein solcher verbleibender Knoten seinen Zustand ändert, wird dies im Mittel genau ein anderer Knoten empfangen.

---

<sup>8</sup> Da das Netzwerk kritisch ist.

---

Da die verbleibenden invertierbaren Funktionen nach ihrer Definition auf jede Änderung eines ihrer Eingänge reagieren, ändert also im Mittel genau ein anderer Knoten in der Folge ebenfalls seinen Zustand, wie es die Definition der Kritikalität fordert.

---

### 1.2.3 Bestimmung des gefrorenen Kerns ausgehend von konstanten Funktionen: Containeralgorithmus

---

Ein eleganter Weg zur Bestimmung des gefrorenen Kerns wurde von [12, 18] vorgeschlagen. Ich werde diesen Algorithmus in Abschnitt 3.3 noch verallgemeinern, um ihn auch bei Netzwerken nutzen zu können, bei denen die Ausgangsgrade der Knoten nicht poissonverteilt sind. Diese Methode beruht auf der Annahme, dass fast alle gefrorenen Knoten bestimmbar sind, ausgehend von Knoten mit konstanter Funktion und iterativer Bestimmung aller weiterer Knoten, die hierdurch bedingt ebenfalls konstant werden. Die grundlegende Idee des Algorithmus ist es, das Netzwerk nicht im Vorhinein festzulegen, sondern seine Struktur erst während der Bestimmung des gefrorenen Kerns anhand der Charakteristik des betrachteten Ensembles festzulegen.

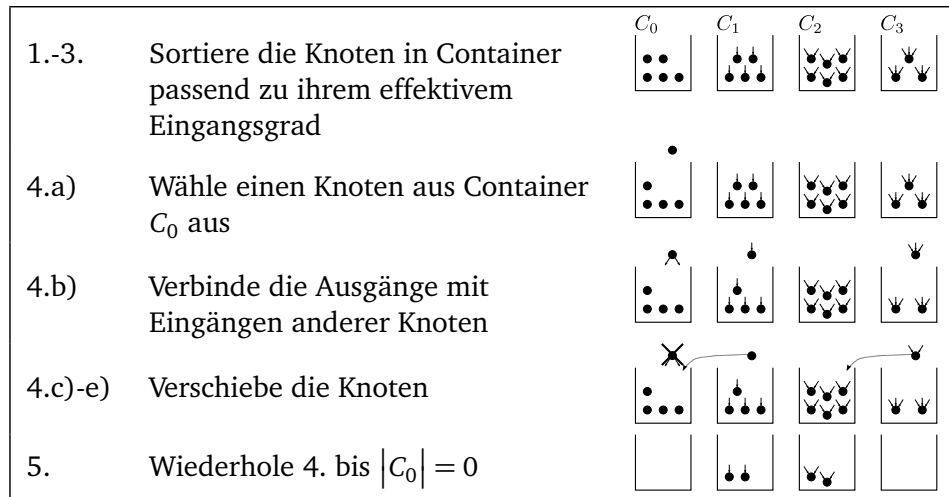
---

#### 1.2.3.1 Algorithmus

---

Der Containeralgorithmus lässt sich als stochastischer Prozess formulieren, bei dem alle Knoten basierend auf ihren Funktionen Containern  $C_i$  zugeordnet werden. In Abbildung 1.3 ist sein Ablauf schematisch dargestellt.

1. Jedem der  $N$  Knoten des Netzwerkes wird eine Aktualisierungsfunktion sowie eine Anzahl von Eingängen anhand der Charakteristik des zugrundeliegenden Ensembles zugeordnet. Um Korrelationen zu vermeiden, werden diese Zuordnungen unabhängig voneinander getroffen.
2. Jeder Knoten wird anhand der Anzahl seiner Eingänge  $i$  einem Container  $C_i$  zugeordnet. Ich werde die Anzahl von Knoten in Container  $C_i$  als  $|C_i| = N_i$  schreiben. Darüber hinaus wird die Anzahl aller Knoten in den Containern als  $N_f = \sum_{i=0}^{i_{\max}} |C_i|$  bezeichnet. All diese Werte werden durch den Algorithmus verändert, im Speziellen wird  $N_f$  in jedem Schritt um 1 erniedrigt.
3. Entscheide für alle  $i > 0$  und für jeden Knoten in Container  $C_i$ , ob dieser eine konstante Funktion hat. Die Wahrscheinlichkeit einer konstanten Funktion mit  $i$  Eingängen bezeichne ich mit  $\beta(i)$ . Falls die Funktion konstant ist, entferne diesen Knoten aus Container  $C_i$ , füge ihn in Container  $C_0$  hinzu.
4. Die folgenden Schritte werden wiederholt bis  $|C_0| = 0$  gilt:
  - a) Wähle einen Knoten  $A$  aus Container  $C_0$ .



**Abbildung 1.3.: Skizze des Containeralgorithmus**

Hier eine grobe Skizze des Containeralgorithmus aus Abschnitt 1.2.3.

- b) Verbinde jeden der  $i$  Eingänge aus allen Knoten in den Containern  $C_i$  mit  $i > 0$  mit der Wahrscheinlichkeit  $\frac{1}{\sum_i N_i}$  mit den Ausgängen von  $A$ .
  - c) Falls genau ein Eingang eines Knotens  $B$  aus Container  $i \geq 1$  zum Knoten  $A$  verbunden wurde, wird dessen Aktualisierungsfunktion mit einer gewissen Wahrscheinlichkeit, die ich als  $\omega_i$  bezeichne, ebenfalls konstant und  $B$  kann in den Container  $C_0$  verschoben werden. Andernfalls wird  $B$  in den Container  $C_{i-1}$  verschoben.
  - d) Falls  $m > 1$  der Eingänge eines Knotens  $B$  aus Container  $i \geq 1$  zum Knoten  $A$  verbunden wurden, wende die vorhergehende Regel  $m$ -mal iterativ an, wobei das  $i$  jedes mal um eins sinkt.
  - e) Entferne den Knoten  $A$  aus dem System. Dies impliziert die Ersetzung  $N_f := N_f - 1$ .
5. Der finale Wert von  $N_f$  entspricht der Anzahl der Knoten, die nicht zum gefrorenen Kern des Netzwerkes gehören, dass durch diesen Algorithmus konstruiert wurde. Die Wahrscheinlichkeitsverteilung von  $N_f$  folgt dem stochastischen Prozess, der diesem Algorithmus entspricht.

Dieser Algorithmus stellt gewisse Anforderungen an die Eigenschaften der Aktualisierungsfunktion und ihrer Verteilung. Wenn ich einen Knoten in Container  $C_i$  betrachte, kann ich diesem nicht mehr seine ursprüngliche Anzahl an Eingängen ansehen. d. h. wenn ich bei einer Funktion mit  $n$  Eingängen  $m$  auf einen konstanten Wert fixiere, muss die Verteilung der eingeschränkten Funktionen identisch derer sein, die von vornherein nur  $n - m$  Eingänge hatten. Bei dem Ensemble der biased Funktionen und dem Ensemble der konstanten und invertierbaren Funktionen ist dies der Fall. Für Letzteres sind alle  $\omega_i = 0$  und damit besonders simpel.

---

### 1.2.3.2 Stochastischer Prozess

---

Dieser Algorithmus lässt sich, wie bereits erwähnt, in einen stochastischen Prozess umformulieren. Hierfür betrachte ich zunächst die deterministischen Differenzengleichungen, die den Prozess für große  $N_i$  gut beschreiben. Die Fluktuationen um diese Mittelwerte können vernachlässigt werden, solange Letztere groß sind. Ich werde mit  $N(t)$  im Folgenden die Gesamtanzahl der Knoten in allen Containern zum Zeitpunkt<sup>9</sup>  $t$  bezeichnen. Im Falle von einem konstanten Eingangsgrad  $K$  für alle Knoten habe ich als Startzustand:

$$N(0) = N \quad (1.7)$$

$$N_0(0) = \beta N \quad (1.8)$$

$$N_i(0) = 0 \quad (1.9)$$

$$N_K(0) = (1 - \beta)N \quad (1.10)$$

In jedem Zeitschritt ändert sich dieses anhand von:

$$\Delta N(t) = -1 \quad (1.11)$$

$$\Delta N_0(t) = -1 + \sum_{i>0} \omega_i (i+1) \frac{N_{i+1}(t)}{N(t)} \quad (1.12)$$

$$\Delta N_i(t) = -i \frac{N_i(t)}{N(t)} + (1 - \omega_i) (i+1) \frac{N_{i+1}(t)}{N(t)} \quad (1.13)$$

$$\Delta N_K(t) = -K \frac{N_K(t)}{N(t)} \quad (1.14)$$

Dies ist nur gültig für genügend große Netzwerke, bei denen die Wahrscheinlichkeit, dass zwei Knoten über mehr als eine gleichgerichtete Kante verbunden sind (Multikanten), verschwindet. Andernfalls müssten sich Knoten sonst im Mittel um mehr als einen Container bewegen, wenn ihre Eingänge einfrieren. Eine Aufsummation der Differenzengleichungen mit großem  $N$  ist in Abbildung 2.3 zu sehen. In der Abbildung fallen negative Werte für  $|C_i|$  auf, obwohl eine Menge nicht weniger als 0 Elemente enthalten kann. Dieses Artefakt rührt daher, dass sich die Differenzengleichungen, im Gegensatz zur echten Dynamik des Algorithmus, auch über die eigentliche Terminierungsbedingung  $|C_0| = 0$  hinweg fortsetzen lassen.

---

<sup>9</sup> Die Zeit  $t$  entspricht nicht der Zeit der echten Dynamik des Systems sondern vielmehr den einzelnen Iterationen des Containeralgorithmus.

Für  $N \rightarrow \infty$  lassen sich diese Differenzengleichungen durch Differenzialgleichungen in den relativen Größen  $n_i := \frac{N_i}{N}$  ersetzen und aufintegrieren:

$$n(0) = 1 \quad (1.15)$$

$$n_0(0) = \beta \quad (1.16)$$

$$n_i(0) = 0 \quad (1.17)$$

$$n_K(0) = 1 - \beta \quad (1.18)$$

$$\frac{dn_0}{dt} = -1 + \sum_{i>0} \omega_i (i+1) n_{i+1} \quad (1.19)$$

$$\frac{dn_i}{dt} = -in_i + (1 - \omega_i) (i+1) n_{i+1} \quad (1.20)$$

$$\frac{dn_K}{dt} = -Kn_K \quad (1.21)$$

Im kritischen Fall gehen alle Container gleichzeitig gegen 0, siehe auch Abbildung 2.3. Dies würde bedeuten, dass in jedem Fall alle Knoten des Netzwerkes einfrieren, was nicht der Realität entspricht. Würde man das Abbruchkriterium modifizieren und bspw. bereits bei  $|C_0| = 1$  terminieren, erhielte man, wie in [12] gezeigt, ebenfalls ein falsches Skalenverhalten.

Für ein realistisches System müssen stochastische Fluktuationen in den Containergrößen  $N_i$  berücksichtigt werden. Durch Einführen zufällige Fluktuationen  $\xi$  mit Mittelwert  $\langle \xi \rangle = 0$  und Varianz  $\sigma_\xi^2 = 1$  in Gleichung (1.19) erhalte ich eine Langevin-Gleichung

$$\frac{dn_0}{dt} = -1 + \sum_{i>0} \omega_i (i+1) n_{i+1} + \xi, \quad (1.22)$$

die (1.19) ersetzt. Die Lösung der Langevin-Gleichung (1.15)-(1.18),(1.22), (1.20),(1.21) mittels Ableitung einer Fokker-Planck-Gleichung führt zum korrekten Skalenverhalten für die Anzahl der nicht-gefrorenen Knoten. Dies zeigt die Bedeutung der Fluktuationen in dem betrachteten Prozess.

---

## 2 Entstehung des gefrorenen Kerns in kritischen Booleschen Netzwerken

Der Inhalt dieses Kapitels beruht auf meiner gemeinsamen Arbeit mit Barbara Drossel, die bereits in [26] veröffentlicht wurde.

---

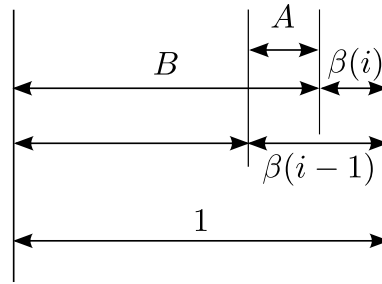
### 2.1 Einleitung

---

In der Einleitung (Abschnitt 1.2.2.1) habe ich bereits das typische Skalenverhalten von relevanten und nicht-gefrorenen Knoten kurz beschrieben. Zur Bestimmung des gefrorenen Kerns wurde in [12, 18] ein effizienter Algorithmus vorgeschlagen, den ich in Abschnitt 1.2.3 vorgestellt habe. Dieser beginnt mit allen Knoten, die eine konstante Aktualisierungsfunktion haben, und bestimmt iterativ alle weiteren Knoten, die aufgrund der konstanten Knoten ebenfalls einen konstanten Wert annehmen. Dieser Algorithmus ermöglicht es, numerisch sehr große Netzwerke (bei mir bis zu  $N = 2^{32}$ ) zu untersuchen, welche zu groß für eine direkte Simulation der Dynamik sind. Darüber hinaus kann dieser Algorithmus in einen stochastischen Prozess übersetzt werden. Aus der Fokker-Planck-Gleichung dieses Prozesses lassen sich die in Abschnitt 1.2.2.1 erwähnten Skalengesetze und viele weitere analytisch ableiten. Außerdem erklärt dieser Ansatz sehr anschaulich, warum, von  $\sim N^{\frac{2}{3}}$  Knoten abgesehen, jeder Knoten für alle Anfangsbedingungen immer auf denselben Wert einfriert. Die Anzahl von Knoten, die nur auf manchen Attraktoren auf einen gewissen fixen Wert einfrieren, aber nicht auf allen, skaliert ebenfalls mit  $N^{\frac{2}{3}}$  und wird so zu einem verschwindend geringen Anteil im thermodynamischen Limes für unendlich große Netzwerke.

Ich werde im Folgenden zeigen, dass die Annahme, der gefrorene Kern könne ausgehend von Knoten mit konstanter Funktion gewonnen werden, für genügend große Werte von  $K$  falsch ist. Bei biased Funktionen genügt bereits ein Wert von  $K$  größer 4. Aus diesem Grund habe ich den Entstehungsprozess des gefrorenen Kerns weitergehend untersucht. Es hat sich gezeigt, dass der gefrorene Kern für große und sehr kleine Werte von  $K$  sehr ähnliche Eigenschaften hat, bis auf den Tatsache, dass er sich nicht immer ausgehend von konstanten Funktionen bestimmen lässt. Hierfür habe ich sowohl Computersimulationen durchgeführt als auch analytische Überlegungen angestellt.

Dieses Kapitel ist folgendermaßen gegliedert: In Abschnitt 2.2 werde ich anhand von numerischen und analytischen Untersuchungen das Problem der



**Abbildung 2.1.: Visualisierung der Wahrscheinlichkeitskomponenten aus Gleichung (2.2)**

Mit  $A = \beta(i-1) - \beta(i)$  und  $B = 1 - \beta(i)$  sowie  $\omega_i = \frac{A}{B}$ .

Containermethode beleuchten. Darauf aufbauend suche ich in Abschnitt 2.3 nach qualitativen Unterschieden im Einfrierverhalten für Netzwerke mit  $K > 4$  und  $K \leq 4$ . In Abschnitt 2.4 präsentiere ich eine Mean-Field-Überlegung für den Entstehungsprozess des gefrorenen Kerns, die einen alternativen Blick auf den Einfrierprozess liefert. Abschließend werden die Ergebnisse in Abschnitt 2.5 zusammengefasst und diskutiert.

## 2.2 Modell

In diesem Kapitel werde ich nur Netzwerke mit einer konstanten Eingangsgradverteilung und Poisson-Ausgangsgradverteilung betrachten. Der konstante Eingangsgrad wird im Folgenden mit  $K$  bezeichnet. Es genügt, sich auf diesen einfachen Fall zu beschränken, da sich bereits hier ein Mangel im Containeralgorithmus offenbart.

Ich werde in diesem Kapitel, wenn nicht anders erwähnt, von biased Aktualisierungsfunktionen ausgehen. Eine solche Funktion mit  $K$  Eingängen entspricht mit der Wahrscheinlichkeit

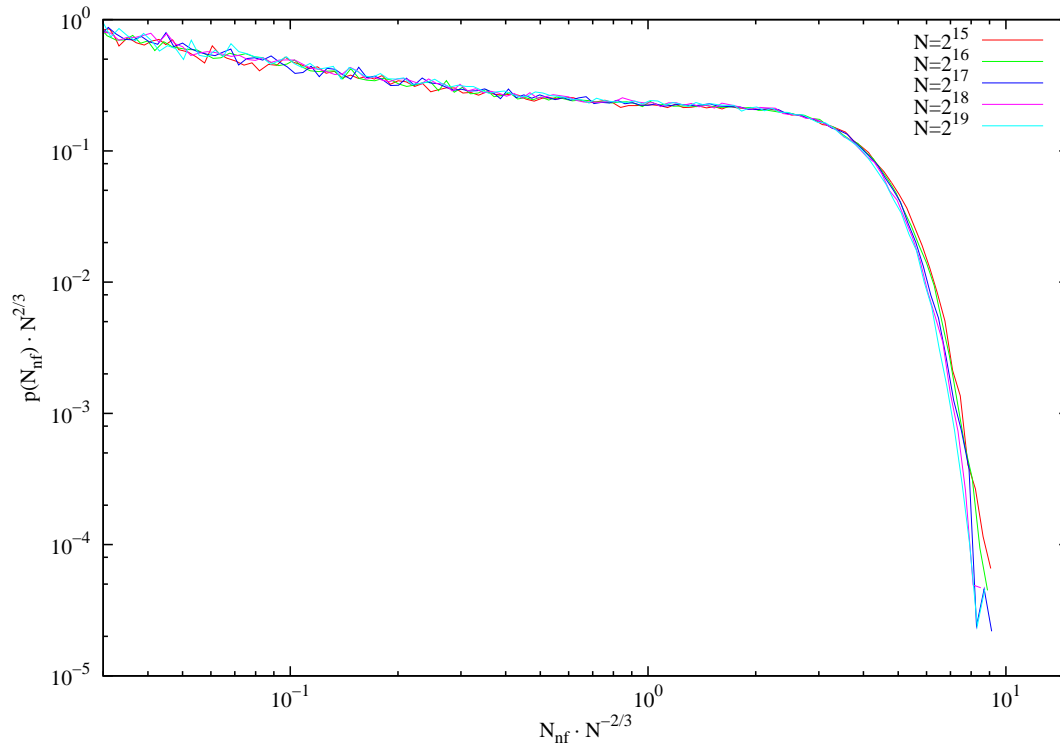
$$\beta(k) = p^{2^K} + (1 - p)^{2^K} \quad (2.1)$$

einer konstanten Funktion.  $\beta(k)$  ist zugleich ein wichtiger Parameter des Containeralgorithmus. Des Weiteren benötige ich die Wahrscheinlichkeit, dass eine nicht konstante biased Funktion mit  $i$  Eingängen nach dem Fixieren eines Einganges zu einer konstanten Funktion wird, welche gegeben ist durch

$$\omega_i = \frac{\beta(i-1) - \beta(i)}{1 - \beta(i)}. \quad (2.2)$$

In Abbildung 2.1 sind die einzelnen Wahrscheinlichkeitskomponenten visualisiert.  $A$  entspricht dem Zähler und  $B$  dem Nenner von Gleichung (2.2). Aufgrund der Voraussetzung weiß man, dass die Funktion nicht-konstant bei  $i$  Eingängen war, wofür die Wahrscheinlichkeit bei  $1 - \beta(i) = B$  liegt. Eine Funktion mit  $i-1$  Eingängen ist mit der Wahrscheinlichkeit  $\beta(i-1)$  konstant,





**Abbildung 2.2.: Wahrscheinlichkeitsverteilung der Anzahl der nicht-gefrorenen Knoten  $N_{\text{nf}}$  für  $K = 4$**

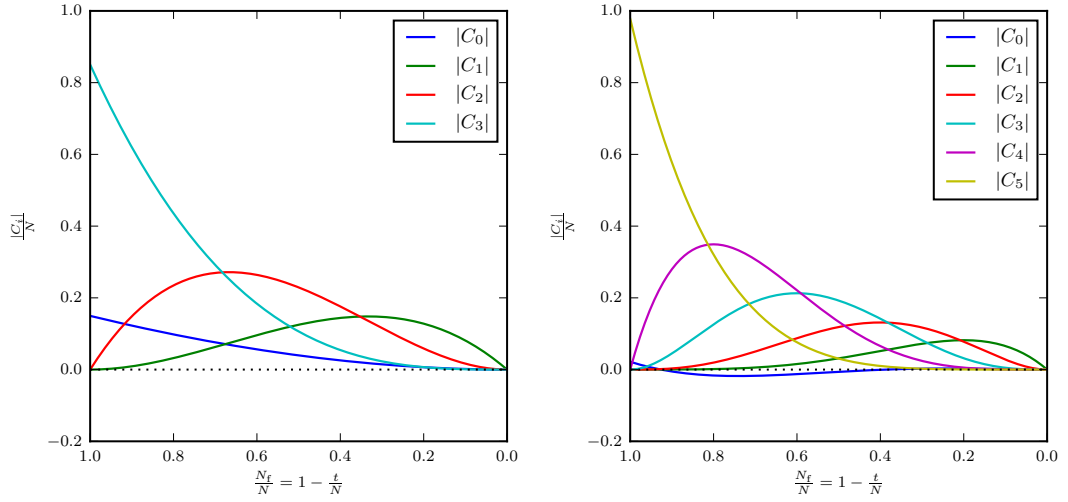
Für verschiedene Werte von  $N \in \{2^{15}, \dots, 2^{19}\}$ , skaliert mit  $N^{\frac{2}{3}}$ . Die Daten wurden mittels der Containermethode generiert, die den gefrorenen Kern iterativ ausgehend von Knoten mit konstanten Funktionen bestimmt. Es wurden insgesamt  $10^5$  zufällige Netzwerke untersucht.

wobei hiervon nach Voraussetzung die Wahrscheinlichkeit, dass die Funktion bei  $i$  Eingängen konstant war, abgezogen werden muss:  $\beta(i-1) - \beta(i) = A$ .

### 2.2.1 Typisches Ergebnis der Containermethode

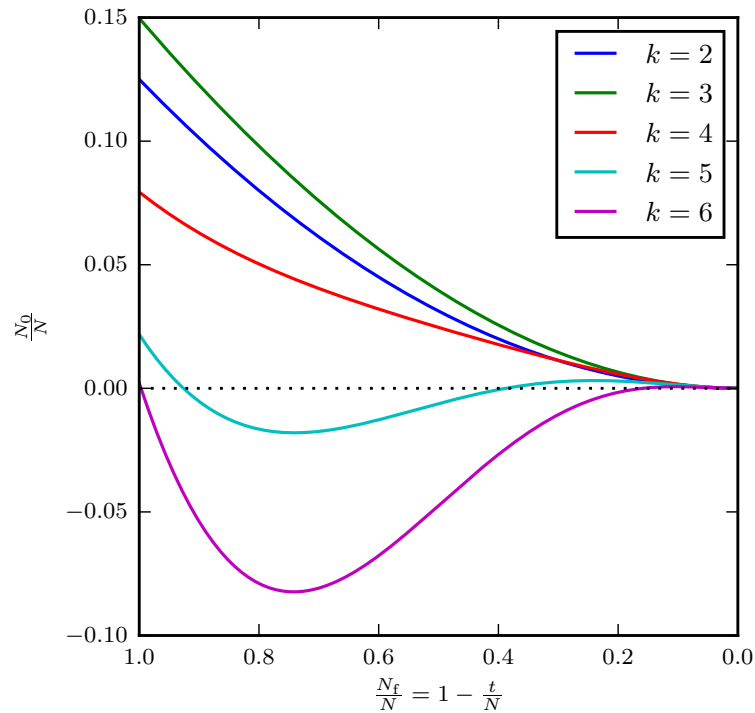
In Abbildung 2.2 ist die Wahrscheinlichkeitsverteilung der Anzahl der nicht-gefrorenen Knoten für  $K = 4$  gezeigt, die mittels der Containermethode bestimmt wurde. Die Daten für unterschiedliche Werte von  $N$  wurden durch das Skalieren mit  $N^{\frac{2}{3}}$  zum Zusammenfallen zu einer universellen Kurve gebracht. In Übereinstimmung mit der allgemeinen Theorie aus [12, 18] ist diese Skalierungsfunktion unabhängig von  $K$  und identisch zu derjenigen, die für  $K = 2$  in [12] vorgestellt wurde.

Wenn dieser Algorithmus jedoch für Werte  $K \geq 5$  angewandt wird, schlägt er fehl: Nur ein sehr kleiner Teil der Knoten wird als gefroren detektiert, wenn man ausschließlich von Knoten mit konstanter Funktion ausgeht.



**Abbildung 2.3.: Größe der Container über die Zeit**

Erhalten durch Iterieren der deterministischen Gleichungen (1.7-1.14) für Netzwerke mit  $N = 10'000$  Knoten und Eingangsgrad von  $K = 3$  (links) bzw.  $K = 5$  (rechts). In der rechten Abbildung wird  $|C_0|$  für genügend große  $N_{\text{nf}}$  noch einmal positiv. Nichtphysikalische  $|C_i| < 0$  entstehen durch das Ignorieren der Terminierungsbedingung  $|C_0| = 0$  des Containeralgorithmus.



**Abbildung 2.4.: Anzahl der Knoten  $N_0$  in Container  $C_0$**

Erhalten durch Iterieren der deterministischen Gleichungen (1.7-1.14) für unterschiedliche Werte von  $K$  und identischer Netzwerkgröße  $N = 10'000$ . Für Werte von  $K \geq 5$  kreuzt  $N_0$  die x-Achse bereits zu einem Zeitpunkt  $t < N$ . Nichtphysikalische  $N_0 < 0$  entstehen durch das Ignorieren der Terminierungsbedingung  $|C_0| = 0$  des Containeralgorithmus.

---

## 2.2.2 Deterministisches Modell für die Containergrößen

---

Um das Versagen der Containermethode für Werte von  $K \geq 5$  zu verstehen, kann ich auf die Differenzengleichungen aus Abschnitt 1.2.3.2 zurückgreifen. In Abbildung 2.3 sind diese Gleichungen aufintegriert. Wie man sieht, gehen für  $K = 3$  alle Containergrößen gleichzeitig gegen 0, während für  $K = 5$  der Container  $C_0$  verfrüht leer wird und somit der Algorithmus vorzeitig terminieren würde.

Abbildung 2.4 zeigt die Anzahl der Knoten in Container  $C_0$  für verschiedene Werte von  $K$ , gewonnen mittels einer numerischen Iteration der deterministischen Differenzengleichungen. Wie man sieht, fällt  $N_0$  monoton von seinem Anfangswert zum Zeitpunkt  $t = 0$  bis zu 0 zum Zeitpunkt  $t = N$  für  $K < 5$  ab. Für  $K \geq 5$  wird  $N_0$  bereits für einen Wert von  $\frac{t}{N}$  nahe 0 negativ und wird erst wieder positiv für  $\frac{t}{N}$  nahe 1. Hier erreicht es ein lokales Maximum und fällt wieder auf 0 zum Zeitpunkt  $t = N$ .

Die Tatsache, dass  $N_0$  negativ wird, bedeutet, dass der gefrorene Kern nicht ausgehend von den konstanten Funktionen gebildet werden kann. Der Fakt, dass  $N_0$  zu einem späteren Zeitpunkt  $\frac{t}{N}$  wieder positiv wird, bedeutet darüber hinaus, dass es dennoch eine Gruppe Knoten gibt, die gefroren bleibt, sobald erst einmal alle gefroren sind.

Wenn eine allgemeinere Klasse von Aktualisierungsfunktionen anstelle der verwendeten biased Funktionen gewählt werden würde, könnte die Bedingung  $N_0(t) = 0$  bereits für  $K = 3$  eine nicht-triviale ( $t < N$ ) Lösung haben. Im Allgemeinen wird das Ensemble der Aktualisierungsfunktionen aus Sicht des Containeralgorithmus durch Wahrscheinlichkeiten  $\beta_i$  und  $\omega_i$  charakterisiert, die nicht unbedingt Gleichung (2.2) der biased Funktionen erfüllen müssen. Für  $K = 3$  gibt es folgende Bedingung an die Existenz einer Lösung  $N_0(t) = 0$  für  $\frac{t}{N} < 1$ :

$$0 < \omega_2 < \frac{1}{3} \tag{2.3}$$

$$\frac{1}{2} < \omega_1 < 1 - \frac{1}{3(1 - \omega_2)} \tag{2.4}$$

$$\beta = \frac{1}{3(1 - \omega_1)(1 - \omega_2)}. \tag{2.5}$$

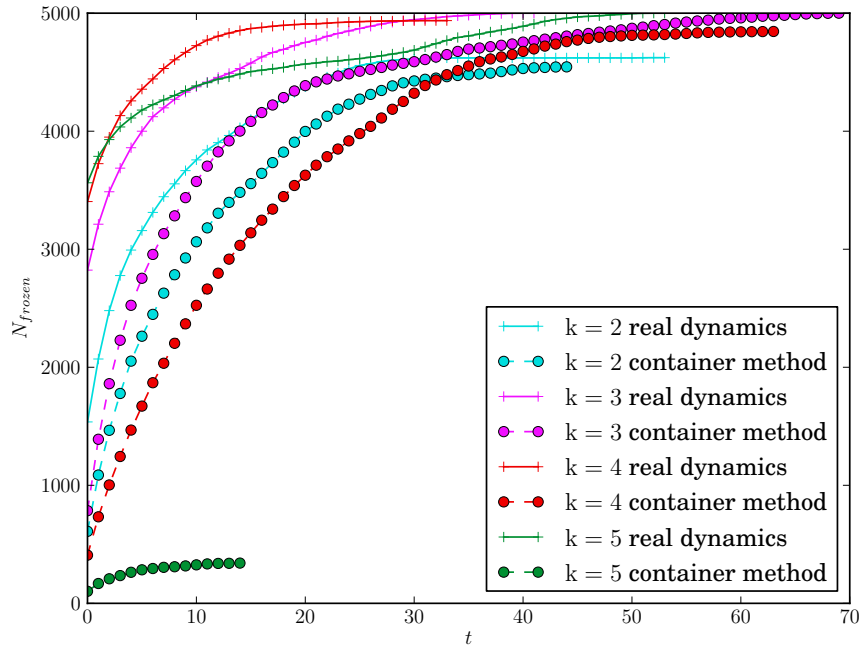
Gleichung (2.5) ist die Bedingung für die Kritikalität des Netzwerkes [18].

---

## 2.2.3 Beispielhafter Vergleich des Containeralgorithmus mit der echten Dynamik

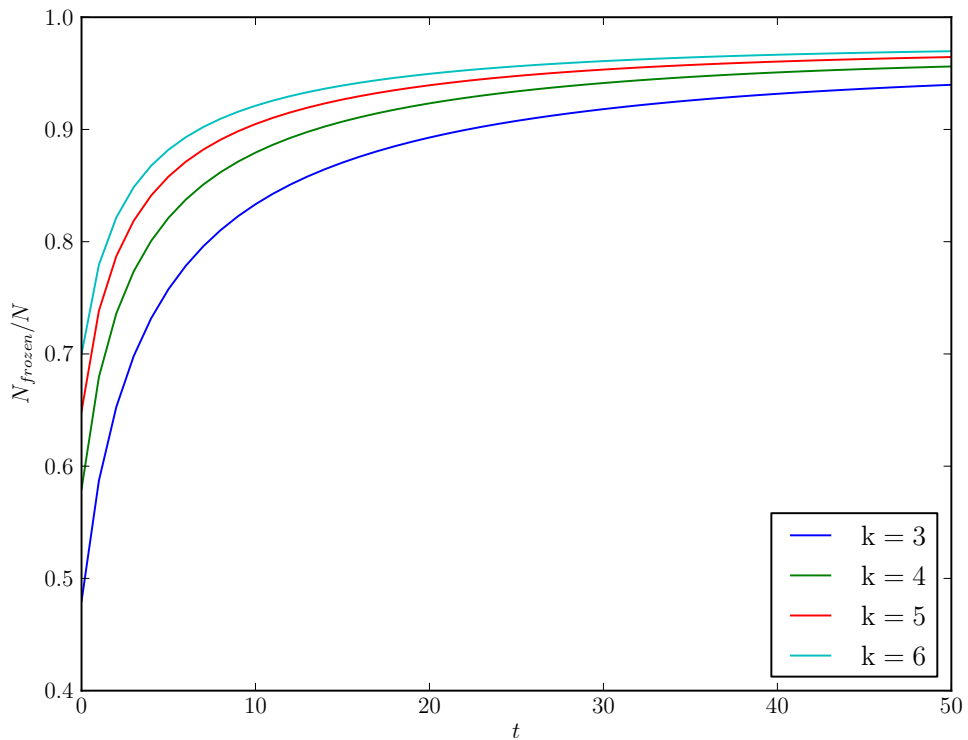
---

Bisher habe ich mich auf die Größe des gefrorenen Kerns konzentriert. Für  $K < 5$  und biased Funktionen lässt sich dieser ausgehend von konstanten Funktionen iterativ bestimmen. Allerdings spiegelt die Containermethode nicht die echte Dynamik der Entstehung des gefrorenen Kerns wider. Um



**Abbildung 2.5.: Einfriedynamik der echten Dynamik verglichen mit dem Containeralgorithmus**

Gezeigt sind 4 zufällige Netzwerke mit  $K \in \{2, \dots, 5\}$  und  $N = 5'000$ . Im Fall der echten Dynamik wurde hier ein Knoten als gefroren angenommen, sobald er seinen Zustand für die restliche Simulationsdauer nicht mehr änderte.



**Abbildung 2.6.: Anteil der gefrorenen Knoten als Funktion der Zeit**

Für unterschiedliche Werte von  $N$ . Gemittelt über mehrere 1'000 Netzwerke mit  $2^{14}$  Knoten, ausgehend von einem zufälligen Startzustand.

---

diese Diskrepanz zu verstehen, habe ich Computersimulationen der echten Dynamik der Netzwerke durchgeführt. Um den Einfluss der Knoten mit konstanter Aktualisierungsfunktion auf die Einfrierdynamik zu untersuchen, habe ich die Anzahl der von einfachen Computersimulationen als gefroren gefundenen Knoten der Netzwerke mit der Anzahl der durch das Fixieren ihrer Eingänge einfrierenden Knoten verglichen. Letzteres entspricht exakt der Containermethode, allerdings mit einer anderen Zeitachse. In den Simulationen hierzu habe ich jene Knoten vom Zeitpunkt ihrer letzten Änderung an als gefroren angenommen, die ihren Zustand in der verbleibenden Simulationsdauer nicht mehr geändert haben. Die Menge der gefrorenen Knoten ist im Allgemeinen größer als der gefrorene Kern, der durch die Containermethode bestimmt wird. Trotzdem lässt sich hier bereits eine erste Tendenz ablesen.

Abbildung 2.5 zeigt das Ergebnis eines solchen Vergleiches für  $K = 2, 3, 4, 5$ . Für  $K = 2, 3, 4$  ist die finale Menge der gefrorenen Knoten bei beiden Methoden in etwa gleich. Dies gilt allerdings nicht für frühe Zeitpunkte im Prozess. Hier ist die Anzahl der durch konstante Funktionen gefrorenen Knoten wesentlich kleiner als die Anzahl der Knoten die ihren Zustand nicht mehr ändern werden. Der Unterschied der beiden Methoden wächst mit  $K$  an. Für  $K = 5$  weichen die beiden Methoden erheblich voneinander ab.

---

## 2.3 Computersimulationen des Entstehungsprozesses des gefrorenen Kerns

---

Im Folgenden ist es mein Ziel, ein besseres Verständnis der eigentlichen Dynamik, die zum Entstehen des gefrorenen Kerns führt, zu ermöglichen. Insbesondere werde ich untersuchen, ob es qualitative Unterschiede zwischen Netzwerken gibt, bei denen die Containermethode anwendbar ist, und solchen, bei denen sie scheitert. Für die Festlegung auf konstante Eingangsgradverteilung und biased Funktionen liegt diese Schwelle bei  $K < 5$  und  $K \geq 5$ .

In diesem Abschnitt präsentiere ich Simulationsergebnisse für die echte Dynamik von Booleschen Netzwerken. Insbesondere liegt das Augenmerk auf qualitativen Änderungen im Entstehungsprozess des gefrorenen Kerns in Abhängigkeit vom mittleren Grad  $k$ .

---

### 2.3.1 Einfrierdauer

---

In Abbildung 2.6 wird der Anteil der gefrorenen Knoten für verschiedene Werte von  $K$  als Funktion der Zeit dargestellt. Jede Kurve ist gemittelt über mehrere 1'000 Netzwerke. Jedes Netzwerk hatte  $N = 2^{14}$  Knoten. Für größere Werte von  $N$  haben sich diese Kurven nicht mehr stark verändert. Simuliert wurde, bis sich ein Attraktor gefunden hatte. Knoten, die sich auf diesem nicht mehr verändert haben, wurden als gefroren angenommen. Falls bis zum Ende der Simulation  $t_{\max} = \frac{N}{2}$  kein Attraktor erreicht wurde, habe ich

---

angenommen, dass alle Knoten zu exakt diesem Zeitpunkt einfrieren. Es wird in meinen Statistiken also keine Transiente und keinen Attraktor geben, der länger als  $t_{\max}$  ist. Des Weiteren werden Transienten der Länge  $t_{\max} - 1$  und Attraktoren der Länge 1 (Fixpunkte) unnatürlich häufig vorkommen. Diese Annahme führt demnach zu einem kleinem finite-size Effekt, der aber meine Ergebnisse nur sehr wenig für die von mir simulierten Werte von  $N$  verfälscht.

Je größer die Werte von  $K$ , desto schneller läuft das Einfrieren ab. Allerdings lässt sich kein qualitativer Unterschied in den Kurven für  $K < 5$  und  $K \geq 5$  erkennen.

---

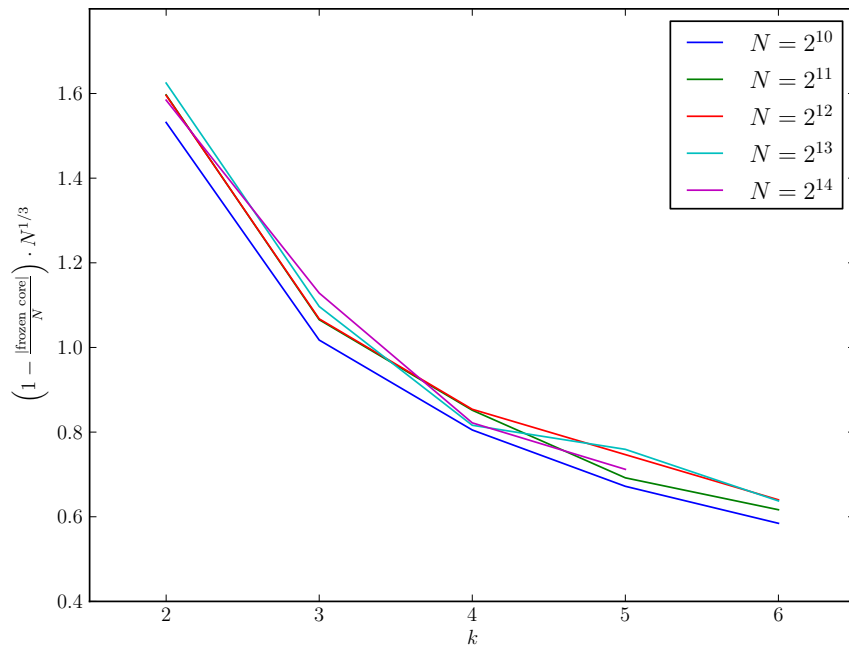
### 2.3.2 Gefrorene Knoten auf unterschiedlichen Attraktoren

---

Als nächstes habe ich untersucht, ob auf unterschiedlichen Attraktoren die gleichen Knoten einfrieren, indem ich die Anzahl der Knoten ausgewertet habe, die auf allen gefundenen Attraktoren eingefroren sind. Hierfür habe ich die Simulationen mehrere Male mit identischem Netzwerk, aber zufälligem Anfangszustand gestartet. Für  $K \geq 5$  lässt sich der gefrorene Kern nicht mehr ausgehend von den konstanten Funktionen bestimmen. Daher bestand die Möglichkeit, dass die gefrorenen Knoten auf verschiedenen Attraktoren unterschiedliche Zustände annehmen. Laut Definition des gefrorenen Kerns (siehe Abschnitt 1.2.1.2) wären sie damit nicht mehr in diesem enthalten. Daher habe ich geprüft, ob gefrorene Knoten immer auf den gleichen Wert einfrieren und ob der Anteil der auf den Wert 1 eingefroren Knoten dem Wert von  $p$  entspricht.

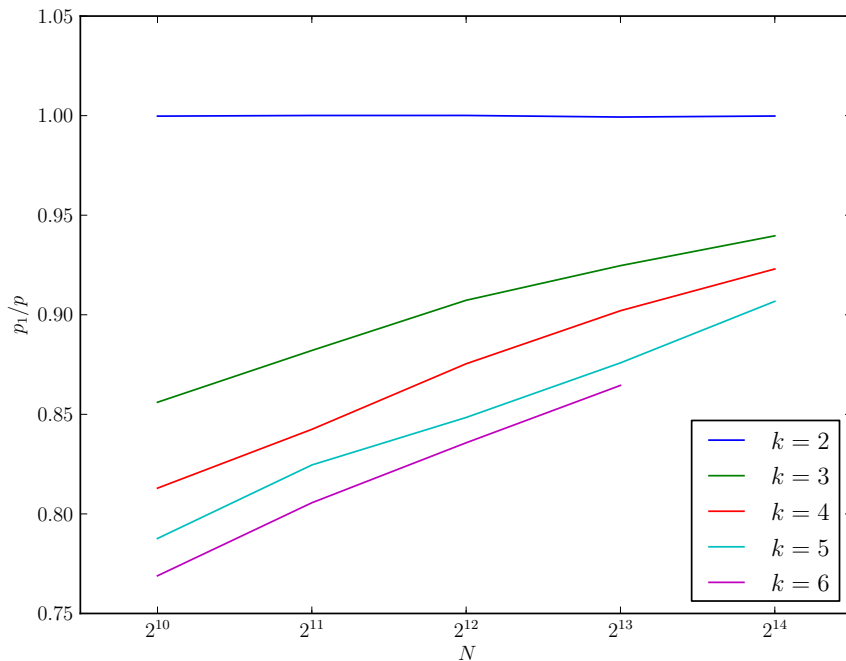
Abbildung 2.7 zeigt den Anteil der Knoten, die nicht für alle 200 getesteten Anfangszustände auf den gleichen Wert einfrieren und somit nicht zum gefrorenen Kern gehören. Es wurde über mindestens einige hundert Netzwerke mit unterschiedlicher Knotenanzahl  $N$  gemittelt. Die Daten sind mit  $N^{\frac{1}{3}}$  skaliert, um eine universelle Kurve zu erhalten. Die Kurven für unterschiedliche  $N$  stimmen sehr gut überein. Dies impliziert, dass nur ein Anteil von höchstens  $\sim N^{-\frac{1}{3}}$  aller Knoten nicht für sämtliche Anfangsbedingungen einfriert bzw. nicht immer auf den gleichen Wert.

Der Anteil  $p_1$  der Knoten, die auf den Wert 1 einfrieren, wird in Abbildung 2.8 normiert bezüglich  $p$  dargestellt. Das Verhältnis  $\frac{p_1}{p}$  nähert sich 1 mit wachsender Knotenanzahl  $N$  von unten an. Die Daten zeigen darüber hinaus, dass es wahrscheinlicher wird, dass ein Knoten auf das Mehrheitsbit 1 einfriert, wenn das Netzwerk kleiner und  $K$  größer ist. Der Grund hierfür ist vermutlich, dass Netzwerke mit größerem  $K$  und kleinerem  $N$  mehr kurze Verbindungsschleifen von einem Knoten zu sich selbst enthalten. Solche Schleifen sollten die Wahrscheinlichkeit erhöhen, dass ein Knoten auf seinem Mehrheitsbit einfriert.



**Abbildung 2.7.: Anteil der Knoten, die nicht für verschiedene Anfangsbedingungen auf den gleichen Wert einfrieren**

Diese Knoten sind folglich nicht Teil des gefrorenen Kerns und ggf. noch nicht einmal gefroren. Gemittelt wurde über mehrere hundert Netzwerke mit unterschiedlicher Knotenanzahl  $N$ . Es wurden mindestens 200 Anfangsbedingungen je Netzwerk getestet. Die Daten wurden jeweils mit  $N^{\frac{1}{3}}$  skaliert. Der Wert für  $N = 2^{14}$  und  $K = 5$  wurde aufgrund langer Rechenzeit nicht berechnet.



**Abbildung 2.8.: Anteil  $p_1$  der Knoten, die auf den Wert 1 einfrieren**

Normiert mittels  $p$ . Gemittelt über mehrere hundert Netzwerke für verschiedene Knotenanzahlen  $N$ . Es wurden mindestens 200 Anfangsbedingungen je Netzwerk getestet. Für größere Netzwerke bzw. kleinere mittlere Grade geht  $p_1$  gegen  $p$ .

---

### 2.3.3 Eigenschaften der Transiente

---

Um die Einfriergeschwindigkeit und ihre Fluktuationen genauer zu verstehen, habe ich für unterschiedliche Netzwerke jeweils untersucht, wie lange es dauert, bis Knoten zum letzten Mal ihren Zustand ändern (d. h. wann sie einfrieren), und wie stark diese Zeit für unterschiedliche Startzustände des Netzwerkes variiert. In Abbildung 2.9 sind die Ergebnisse dieser Rechnung gezeigt. Es fällt unmittelbar ein sehr markantes Muster der oberen Schranke für diese Werte auf (Hüllkurve). Dieses Muster lässt sich jedoch sehr simpel durch statistische Eigenschaften von ganzen Zahlen begründen. Die Zeiten vom Startzustand bis zum Einfrieren sind jeweils ganze Zahlen, und zwar jedes Mal gleich viele mit einer festen oberen Schranke für jede einzelne Zahl.

Wenn der Mittelwert

$$\langle n \rangle = \frac{1}{m} \sum_{i=1}^m n_i$$

gegenüber der Varianz

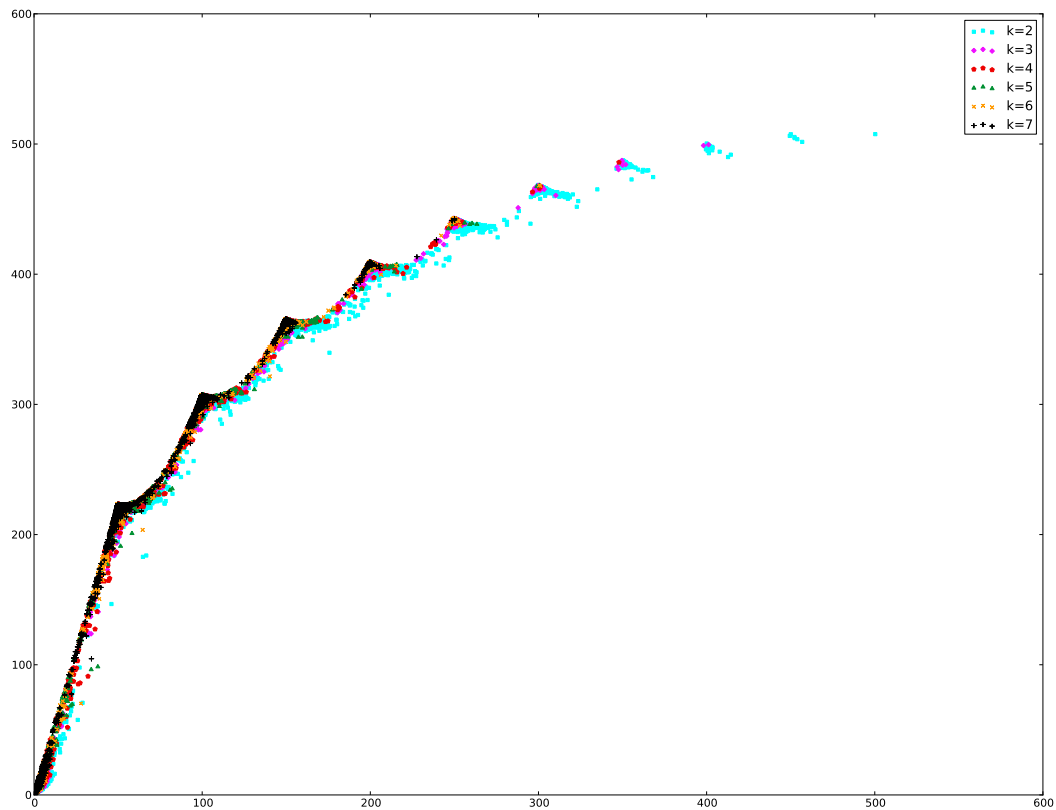
$$\sigma_n = \sqrt{\frac{1}{m} \sum_{i=1}^m (n_i - \langle n \rangle)^2}$$

einer gegebenen Anzahl  $m$  von ganzen Zahlen  $n_1, \dots, n_m \in \mathbb{N}$  mit  $n_i \leq M$  aufgetragen wird, erhält man ein markantes Muster wie in Abbildung 2.9, unabhängig vom untersuchten Gegenstand. Es lassen sich sogar die Anzahl der lokalen Maxima und die initiale Steigung leicht herleiten. Ein lokales Maximum entsteht immer dann, wenn sich die zu mittelnden Werte maximal weit voneinander entfernt befinden. Das  $i$ -te Maximum entsteht genau für die Zahlenfolge (abgesehen von Permutationen)  $\underbrace{0, 0, \dots, 0}_{m-i \text{ mal}}, \underbrace{M, M, \dots, M}_{i \text{ mal}}$ .

Die Hüllkurve von allen Punkten entsteht, indem zusätzlich immer die rechteste 0 kontinuierlich bis auf  $M$  erhöht wird. Aus diesen Zahlenfolgen lassen sich alle lokalen Maxima, die Anfangssteigung sowie die Hülle der Punkte ableiten. Der Parameter  $M$  bestimmt die Skala der x- und y-Achse sowie die Dichte der Punkte. Der Parameter  $m$  hingegen legt die Anzahl  $m - 1$  der lokalen Maxima fest. In Abbildung 2.10 habe ich dies für alle möglichen ganzzahligen Zahlenfolgen in einem Intervall visualisiert. Die Punkte der Berechnung aus Abbildung 2.9 ließen sich mit den Parametern  $M = 1'000$  und  $m = 20$  reproduzieren.

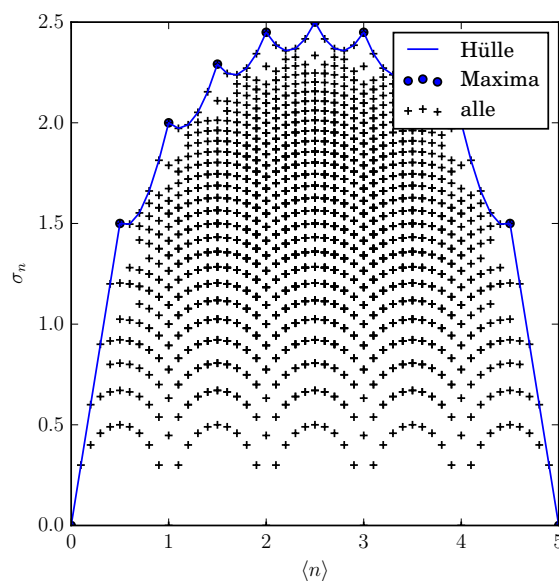
Ich habe diese Berechnung modifiziert wiederholt und dabei nur Knoten berücksichtigt, die zum gefrorenen Kern gehören. Die Daten ähneln dem linken Teil der Abbildung 2.9, bevor das erste lokale Maximum erreicht wird. Auch hier lässt sich kein qualitativer Unterschied zwischen Netzwerken mit  $K < 5$  und  $K \geq 5$  erkennen.





**Abbildung 2.9.: Mittlere Dauer (x-Achse) und Varianz dieser Dauer (y-Achse), bis ein Knoten zum letzte Mal seinen Zustand ändert bei Variation des Startzustandes des Netzwerks**

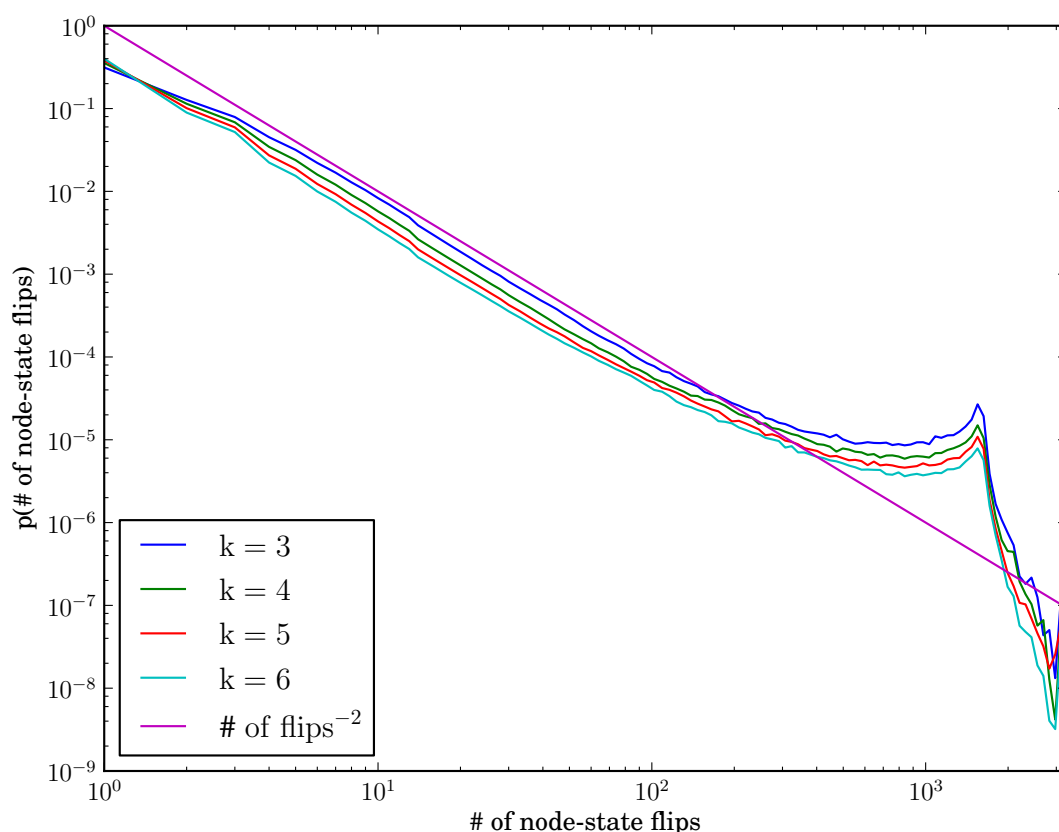
Simuliert wurden jeweils 1'000 Zeitschritte von Netzwerken mit  $N = 1'000$  Knoten. Jeder Punkt entspricht der Statistik eines Knotens über 20 unterschiedliche Startzustände eines Netzwerkes. Ein allgemeines Modell für die Hüllkurve wird in Abbildung 2.10 gezeigt.



**Abbildung 2.10.: Mittelwert und Varianz von ganzen Zahlen**

Mittelwert und Varianz aller möglichen  $n_1, \dots, n_{10} \in \mathbb{N}$  mit  $0 \leq n_i \leq 5$ . Hier erhalte ich eine Hüllkurve wie in Abbildung 2.9, ohne Bezug zum eigentlichen Modell.

Abschließend habe ich ein Histogramm der Anzahl der Zustandswechsel der Knoten auf der Transiente betrachtet. Falls innerhalb der Simulationszeit  $\frac{N}{2}$  kein Attraktor gefunden werden konnte, wurde der letzte Zustand als Fixpunkt angenommen. Abbildung 2.11 zeigt Daten für  $N = 2^{14}$  und unterschiedliche  $K$ . Auch hier konnte kein qualitativer Unterschied für  $K < 5$  und  $K \geq 5$  gefunden werden. Mit wachsendem  $K$  frieren die Knoten früher ein und die Anzahl der Zustandswechsel sinkt. Der Peak am Ende der Kurve ist ein finite-size Effekt, weshalb seine Position mit  $N$  skaliert. Bevor die finite-size Effekte das Verhalten dominieren, scheint die Kurve einem Potenzgesetz mit einem Exponenten  $-2$  zu folgen. Ich werde diesen Exponenten in Abschnitt 2.4 durch eine Mean-Field-Überlegung begründen.



**Abbildung 2.11.: Wahrscheinlichkeitsverteilung für die Anzahl der Zustandswechsel von Knoten auf der Transiente**

Daten wurden gemittelt aus Simulationen mehrere tausend Netzwerke und für eine Knotenanzahl von  $N = 2^{14}$ . Die Schwingungen ab einem Wert von ca. 1000 Wecheln auftreten, sind Artefakte der endlichen Simulationsdauer.

---

## 2.4 Mean-Field-Überlegung für den Entstehungsprozess des gefrorenen Kerns

---

Im Folgenden werde ich eine Mean-Field-Überlegung für den Entstehungsprozess des gefrorenen Kerns durchführen. Diese Berechnung bestimmt die Wahrscheinlichkeit, dass ein Knoten zu einem Zeitschritt seinen Zustand in Abhängigkeit davon wechselt, dass einer seiner Eingänge seinen Zustand im vorherigen Zeitschritt gewechselt hat. Diese Berechnung berücksichtigt keine Korrelationen zwischen Knoten und zwischen Zustandswechseln eines Knotens zu unterschiedlichen Zeitpunkten. Daher kann sie nur für kurze Zeiten, zu denen das Netzwerk den Attraktor noch nicht erreicht hat, Gültigkeit haben. Da die wichtigsten relevanten Schleifen in kritischen RBNs eine Länge in der Größenordnung von  $\sim N^{\frac{1}{3}}$  [12] haben, erwarte ich, dass die finite-size Effekte nach  $\sim N^{\frac{1}{3}}$  Zeitschritten sichtbar werden.

Ich beginne mit einem zufälligen Anfangszustand, bei dem jeder Knoten den Zustand 1 mit Wahrscheinlichkeit  $p$  annimmt. Nach dem ersten Zeitschritt hat jeder Knoten den durch seine Boolesche Aktualisierungsfunktion vorgeschriebenen Zustand, bestimmt durch die Werte an seinen Eingängen. Die Wahrscheinlichkeit, dass dies ein anderer Zustand als zuvor war, ist  $2p(1-p) = F_0$ , d.h. dass der Knoten von 0 zu 1 oder von 1 zu 0 wechselt. In jedem der folgenden Schritte kann ein Knoten genau dann seinen Zustand wechseln, wenn dies auch mindestens einer seiner Eingangsknoten im vorherigen Zeitschritt getan hat. Ich bezeichne mit  $F_t$  die Wahrscheinlichkeit, dass ein Knoten im Zeitschritt  $t$  seinen Zustand ändert. Die Wahrscheinlichkeit, dass mindestens ein Eingang eines Knotens zum Zeitpunkt  $t$  seinen Zustand gewechselt hat, ist  $(1 - (1 - F_t)^K)$ . Hieraus erhalte ich die Gleichung

$$F_{t+1} = (1 - (1 - F_t)^K) F_0. \quad (2.6)$$

Der Fixpunkt dieser rekursiven Gleichung ist durch die Bedingung  $F^* = (1 - (1 - F^*)^K) F_0$  gegeben. Die Gleichung auf der rechten Seite hat ihre maximale Steigung bei  $F^* = 0$ . Diese Steigung ist immer kleiner als 1 wenn  $K$  kleiner ist als  $\frac{1}{2p(1-p)}$ . In diesem Fall konvergiert (2.6) zum Fixpunkt 0, was einem komplett gefrorenen Netzwerk entspricht. Im entgegengesetzten Fall gibt es einen stabilen Fixpunkt bei einer von 0 verschiedenen Zustandswechselrate. An der Grenze zwischen diesen Fällen ist das System kritisch und die Zustandswechselrate konvergiert gegen den Wert 0 ohne diesen zu erreichen. Diese Bedingung für die Kritikalität entspricht der Bedingung in (1.5).

Die Taylorreihe von (2.6) um den Wert  $F_t = 0$  für ein kritisches Netzwerk  $F_0^c = F_0 = 2p(1-p) = \frac{1}{K}$  ergibt:

$$F_{t+1} = \left(1 - (1 - F_t)^K\right) F_0^c \quad (2.7)$$

$$= F_t - \frac{K(K-1)}{2} F_t^2 F_0^c + \mathcal{O}(F_t^3). \quad (2.8)$$

Hieraus erhalte ich

$$\Delta F_{t+1} = F_{t+1} - F_t \approx -\frac{K(K-1)}{2} F_t^2 F_0^c.$$

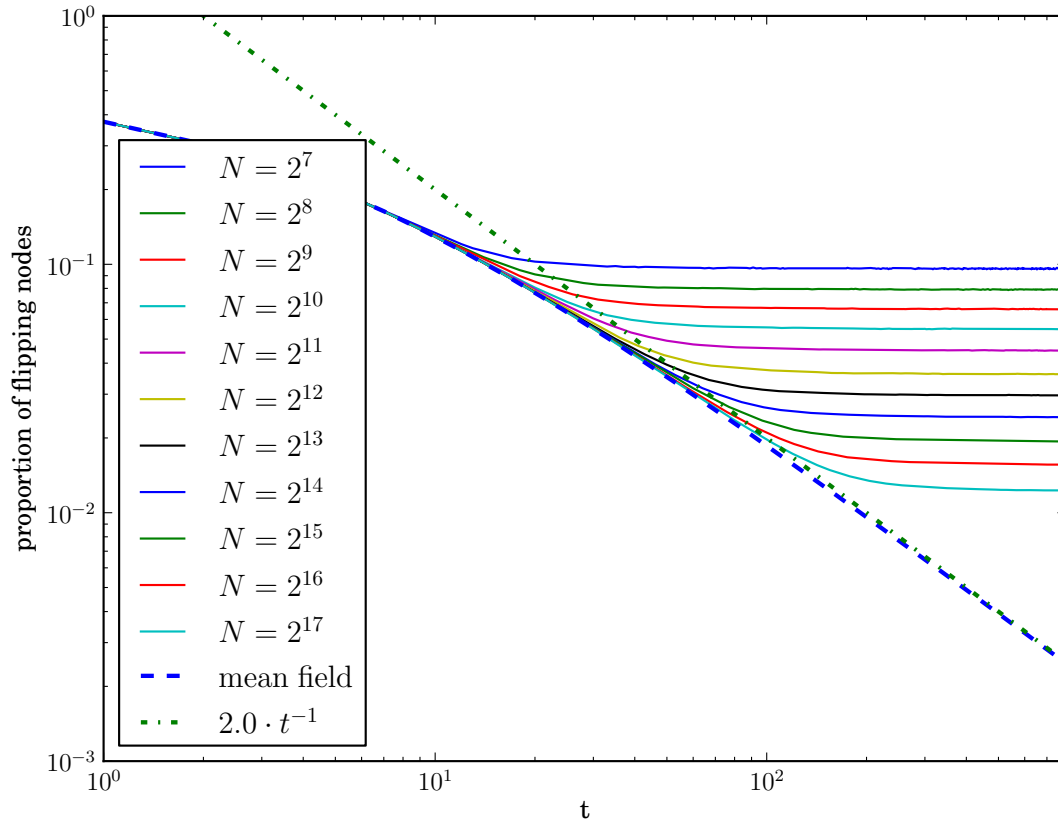
Wenn ich dies in eine Differenzialgleichung übersetze und integriere, ergibt sich

$$F(t) = \frac{2}{(K-1)t}. \quad (2.9)$$

Dies bedeutet, dass die Anzahl der Knoten mit wechselndem Zustand für große Zeiten wie  $\frac{1}{t}$  abnimmt. Die Anzahl der Knoten die zum Zeitpunkt  $t$  zum letzten Mal ihren Zustand wechseln ist proportional zu  $-\dot{F}(t)$ , was proportional zu  $t^{-2}$  ist. Hierdurch habe ich eine Begründung für den in Abbildung 2.11 entdeckten Exponenten gefunden.

Um die Qualität meiner Mean-Field-Überlegung zu überprüfen, habe ich die Anzahl der Knoten bestimmt, die zu einem Zeitpunkt  $t$  in einem echten kritischen Netzwerk ihren Zustand ändern. Abbildung 2.12 zeigt die gefundenen Daten für unterschiedliche Netzwerkgrößen für  $K = 2$ , gemittelt über 10'000 Netzwerke. Zusätzlich zu den simulierten Daten sind in der Abbildung die Ergebnisse der Mean-Field-Überlegung und ein Potenzgesetz mit Steigung  $-1$  gezeigt.

Wie man sieht, stimmen die simulierten Daten bis zu einem Grenzwert exakt mit den Ergebnissen der Mean-Field-Überlegung überein. Eine Untersuchung des Grenzwertes hat gezeigt, dass dieser mit  $N^{\frac{1}{3}}$  skaliert, was in perfekter Übereinstimmung mit den erwarteten finite-size Effekten vom Anfang dieses Abschnitts ist. Zu diesem Zeitpunkt hat das Netzwerk sein stationäres Aktivitätslevel erreicht.



**Abbildung 2.12.: Vergleich der Mean-Field-Überlegung mit simulierten Daten für die Zustandswechselwahrscheinlichkeit in Abhängigkeit von der Zeit**

Die Abbildung zeigt den Anteil der Knoten, die ihren Zustand wechseln, in Abhängigkeit der Netzwerkgröße  $N$  und der Zeit  $t$ , für  $K = 2$ . Gemittelt wurde jeweils über 10'000 Netzwerke. Zusätzlich zu den simulierten Daten enthält der Graph auch die Ergebnisse aus der Mean-Field-Überlegung und das Potenzgesetz aus Gleichung (2.9) mit Steigung  $-1$ .

---

## 2.5 Zusammenfassung

---

In diesem Kapitel habe ich eine Analyse des Entstehungsprozesses des gefrorenen Kerns von kritischen Booleschen Netzwerken präsentiert. Ich habe im Einzelnen Netzwerke mit biased Funktionen betrachtet und bei diesen die Eigenschaften des gefrorenen Kerns in Abhängigkeit vom mittleren Grad  $K$  untersucht. Dabei fand ich heraus, dass sich für  $K > 4$  der gefrorene Kern nicht ausgehend von den Knoten mit konstanter Funktion bestimmen lässt. Trotzdem ließen sich keine qualitativen Unterschiede zwischen dem gefrorenen Kern von Netzwerken mit  $K \leq 4$  und  $K > 4$  finden. Mittels einer Mean-Field-Überlegung für die Anzahl der Knoten, die zu einem bestimmten Zeitpunkt ihren Zustand wechseln, konnte ich zwei Potenzgesetze bestimmen, die sich in meinen Simulationen wiederfinden. Zudem konnte ich zeigen, dass sich der Entstehungsprozess des gefrorenen Kerns selbst unter Vernachlässigung von Korrelationen zwischen Knoten und aufeinanderfolgenden Zustandswechseln korrekt beschreiben lässt. Durch Simulationen konnte ich darüber hinaus zeigen, dass immer die gleichen Knoten, bis auf einen Anteil, der mit  $N^{-\frac{1}{3}}$  skaliert, auf den gleichen Wert einfrieren. Dieses Verhalten ist somit unabhängig vom Startzustand.

Für  $K \leq 4$  lässt sich dieses Ergebnis ausgehend von Knoten mit konstanter Funktion zeigen. Für  $K > 4$  deutet dieses Ergebnis auf die Existenz von größeren Gruppen sich gegenseitig fixierender Knoten hin.

Aufgrund der Universalität von kritischem Verhalten erwarte ich, dass diese Ergebnisse auch für zufällige Boolesche Netzwerke mit anderen Ensembles für die Aktualisierungsfunktionen Bestand haben, ebenso wie für Netzwerke mit anderen Gradverteilungen, solange das zweite Moment dieser Verteilungen endlich ist [25].

Das folgende Kapitel 3 handelt von Netzen mit skalenfreien Gradverteilungen, die ein divergierendes zweites Moment besitzen. Selbst hier scheitert die Containermethode für mittlere Gerade  $K \gtrsim 4$ . Somit können meine Ergebnisse aus diesem Kapitel teilweise ebenfalls auf solche Netzwerke übertragbar werden, auch wenn in diesem Umfeld andere Skalenexponenten zu finden sind.

---

# 3 Skalengesetze in kritischen Booleschen Zufallsnetzen mit allgemeinen Ein- und Ausgangsgradverteilungen

Der Inhalt dieses Kapitels beruht auf meiner gemeinsamen Arbeit mit Barbara Drossel, die bereits in [27] veröffentlicht wurde.

---

## 3.1 Einleitung

---

In Abschnitt 1.2.2.1 der Einleitung habe ich bereits das typische Skalenverhalten von relevanten und nicht-gefrorenen Knoten kurz beschrieben. Diese lassen sich unter anderem über eine analytische Untersuchung des stochastischen Prozesses der Containermethode [12, 18] herleiten, welche schon in Abschnitt 1.2.3 der Einleitung vorgestellt wurde. Die Methode beruht auf der Annahme, dass sich der gefrorene Kern eines Netzwerkes ausgehend von den Knoten mit konstanter Aktualisierungsfunktion herleiten lässt. Im letzten Kapitel habe ich zudem die Grenzen dieses Algorithmus für große mittlere Knotengrade aufgezeigt. Solange der mittlere Grad nicht zu groß ist, lässt er sich jedoch hervorragend einsetzen. Im Folgenden werden gewisse analytischen Überlegungen zur Containermethode angestellt, bei denen der mittlere Grad keine Rolle zu spielen scheint, da hier nur das (vermeintliche) Ende des Prozesses betrachtet wird.

Der Großteil der aktuellen Forschungsergebnisse zu Booleschen Netzwerken zielt auf Netzwerke mit realistischeren Eigenschaften ab. Insbesondere sind viele natürliche Netzwerke skalenfrei, d. h. dass die Ein- oder Ausgangsgradverteilungen der Knoten einem Potenzgesetz folgen [28]. Typische Exponenten der Potenzgesetze liegen zwischen 2 und 3. Bei einem Exponenten von kleiner 3 divergiert das zweite Moment dieser Verteilung mit wachsender Knotenzahl. Aufgrund ihrer Relevanz für natürliche Systeme wurden skalenfreie Netzwerke bereits von vielen Autoren untersucht. Um die Modelle möglichst einfach zu halten, wurde die Verknüpfungsstruktur unter Berücksichtigung der gegebenen Gradverteilung zufällig gehalten. Die Auswahl der Aktualisierungsfunktionen erfolgt typischerweise ebenfalls zufällig aus einem Ensemble. Es wurde in Simulationen beobachtet, dass bei kritischen skalenfreien Netzwerken die Attraktoren kürzer und gefrorene Knoten häufiger werden als bei konventionellen RBNs mit der gleichen Anzahl an Knoten und Verknüpfungen [29, 30]. Darüber hinaus sind Attraktoren empfindlicher gegenüber Störungen an stark verknüpften Knoten, aber nicht an

---

schwach verknüpften Knoten [31, 30]. Auch evolvieren skalenfreie Boolesche Netzwerke schneller und stetiger hin zu einem Zielmuster an einem *Ausgangsknoten* als das konventionelle RBNs tun [32].

Analytische Ergebnisse beschränken sich meistens auf die Berechnung des Phasendiagramms mittels der annealed Approximation [31, 33, 34]. Ebenso wie auch konventionelle RBNs mit einem festen Eingangsgrad zeigen auch skalenfreie RBNs eine gefrorene, kritische und chaotische Phase in Abhängigkeit ihrer Parameter. Ein Vergleich zwischen den Ergebnissen der annealed Approximation und denen aus direkten Simulationen wurde in [35] durchgeführt. Es fand sich nicht immer eine Übereinstimmung zwischen den beiden Ansätzen. In [24] wurde eine analytische Herleitung der Skalenexponenten für die Anzahl der nicht-gefrorenen Knoten für skalenfreie Eingangsgradverteilungen durchgeführt. Die Werte dieser Exponenten hängen stetig vom Exponenten der Eingangsgradverteilungen ab, falls dieser kleiner als 3 ist. Aus Simulationen von RBNs mit skalenfreien Ausgangsgradverteilungen ist bekannt, dass die Eigenschaften der Attraktoren sich vom Fall der skalenfreien Eingangsgradverteilungen unterscheiden [36].

Im Folgenden werde ich die Skalenexponenten der Anzahl der nicht-gefrorenen Knoten für kritische RBNs mit skalenfreien Ein- und/oder Ausgangsgradverteilungen herleiten. Hierfür werde ich in 3.3 zunächst die in Abschnitt 1.2.3 beschriebene Containermethode, die bereits in der Analyse von skalenfreien Eingangsgradverteilungen zum Einsatz kam [24], entsprechend erweitern. In Abschnitt 3.4 werde ich die Methode dann analytisch untersuchen und als Ergebnis eine große Anzahl an Potenzgesetzen präsentieren, in Abhängigkeit der Ein- und Ausgangsgrad-Exponenten und dem jeweiligen Cut-off der Verteilungen. Für einen meiner Fälle erhalte ich das Potenzgesetz, das in [25] bereits gefunden wurde, wenn auch nicht für den gleichen Fall wie dort. Meine analytischen Ergebnisse habe ich zu großen Teilen mittels Simulationen verifiziert, wie in Abschnitt 3.6 dargelegt. Nach einer Beschreibung der hierfür notwendigen Implementierung in Abschnitt 3.7 werde ich die Ergebnisse in Abschnitt 3.8 noch einmal zusammenfassen.

---

## 3.2 Modell

---

In diesem Kapitel betrachte ich Ensembles kritischer RBNs mit  $N$  Knoten. Als Verteilungen für die Ein- bzw. Ausgangsgrade untersuche ich die in Abschnitt 1.1.1 vorgestellte konstante Gradverteilung bzw. die Poissonverteilung. Wenn im Folgenden von einer konstanten Gradverteilung mit nicht ganzzahligem Grad die Rede ist, werden hierfür die beiden benachbarten ganzzahligen Grade im passendem Verhältnis gemischt. Beispielsweise würden für einen mittleren Grad von 3,4 ein Anteil von 60% der Grade exakt 3 und 40% exakt 4 gewählt werden.



Darüber hinaus möchte ich hier insbesondere skalenfreie Ein- und/oder Ausgangsgradverteilungen betrachten:

$$P_{\text{ein}}(k) \propto k^{-\gamma_{\text{ein}}} \quad (3.1)$$

$$P_{\text{aus}}(k) \propto k^{-\gamma_{\text{aus}}} \quad (3.2)$$

Ich werde nur den Fall ohne Korrelationen zwischen Ein- und Ausgangsgrad eines Knotens betrachten, wobei auf den interessanten Fall  $\gamma_{\text{ein}}, \gamma_{\text{aus}} \in (2, 3)$  ein besonderer Fokus gelegt wird, da hier das zweite Moment der Gradverteilungen divergiert

$$\lim_{N \rightarrow \infty} \sum_{k=0}^{k_{\text{max}}} P(k) k^2 \rightarrow \infty, \quad (3.3)$$

obwohl ihr Mittelwert mit  $K_{\text{ein}}$  bzw.  $K_{\text{aus}}$  wohldefiniert ist. Der Mittelwert der Ein- und Ausgangsgradverteilung muss identisch sein, da jeder Eingang eines Knotens auch mit einem Ausgang eines (typischerweise anderen) Knotens verbunden werden muss. Für  $\gamma_{\text{ein}}, \gamma_{\text{aus}} < 2$  divergiert der Mittelwert (mittlere Grad) der Verteilungen und würde eine unabhängige Wahl von  $K$  und  $N$  unmöglich machen.

Ich habe skalenfreie Ein- und Ausgangsgradverteilungen  $P(k) \propto k^{-\gamma}$  auf zwei Arten erzeugt. Diese führen zu einem unterschiedlichen Skalieren des Cut-offs  $k^{\text{max}}(N) := \max \{k : P(k) > 0\}$  mit der Netzwerkgröße  $N$ . Entweder ziehe ich jeden einzelnen Ein- bzw. Ausgangsgrad  $k_i$  zufällig aus der Verteilung  $P(k)$ , was zu dem Cut-off

$$k^{\text{max}} \propto N \quad (3.4)$$

führt. Alternativ fixiere ich die Anzahl der Knoten mit dem Ein- oder Ausgangsgrad  $k$  als  $c \cdot N \cdot P(k)$  (gerundet auf die nächste ganze Zahl), während die Konstante  $c \gtrapprox 1$  so angepasst wird, dass die Stichprobengröße so nahe wie möglich bei  $N$  liegt. Dies führt zu einem Cut-off von

$$k^{\text{max}} \propto N^{\frac{1}{\gamma}}. \quad (3.5)$$

Ich habe den minimalen Wert von  $k$  in skalenfreien Verteilungen auf 2 gesetzt, da dies zu mehr Knoten mit großem Grad  $k$  führt und dadurch das asymptotische Verhalten mit wachsendem  $N$  schneller erreicht wird.

Da wie eben beschrieben erzeugte Stichproben z.B. durch Rundungen nicht mehr exakt der Verteilung entsprechen, stimmt ihr Mittel nicht mit dem Mittelwert der Verteilung überein. Insbesondere das Abschneiden der Verteilung bei  $k^{\text{max}}$  sorgt für ein deutliches Absinken des Mittels. Um dies zu kompensieren, habe ich die Anzahl der Knoten mit  $k = 2$  Verbindungen abweichend von der Verteilung so variiert, dass sich der mittlere Grad  $K$  nicht mit  $N$  ändert, und somit von Anfang an dem asymptotischen Wert

---

entspricht, der im Grenzwert  $N \rightarrow \infty$  angenommen wird. Dieses Vorgehen reduziert somit finize-size Effekte in meinen Simulationen erheblich.

Um die RBNs im betrachteten Ensemble kritisch werden zu lassen, habe ich die Parameter des verwendeten Funktionenensembles jeweils entsprechend dem mittleren Grad  $K$  gewählt. Für solche kritischen Netzwerke wächst die Anzahl der gefrorenen Knoten sublinear mit der Netzwerkgröße  $N$ , das genaue Potenzgesetz wird im Folgenden bestimmt.

---

### 3.3 Verallgemeinerung der Containermethode

---

Eine effiziente Methode zur Bestimmung des gefrorenen Kerns wurde von [12, 18] vorgeschlagen und in Abschnitt 1.2.3 bereits vorgestellt. Ich werde diesen Algorithmus hier etwas verallgemeinern, um ihn auch bei Netzwerken nutzen zu können, deren Ausgangsgrade der Knoten nicht poissonverteilt sind. Die Methode beruht auf der Annahme, dass fast alle gefrorenen Knoten bestimmbar sind, wenn man ausgehend von Knoten mit konstanter Funktion iterativ alle weiteren Knoten, die hierdurch ebenfalls konstant werden, findet. Die grundlegende Idee des Algorithmus ist es, das Netzwerk nicht im Vorherein festzulegen, sondern seine Struktur erst während der Bestimmung des gefrorenen Kerns anhand der Charakteristik des betrachteten Ensembles zu definieren.

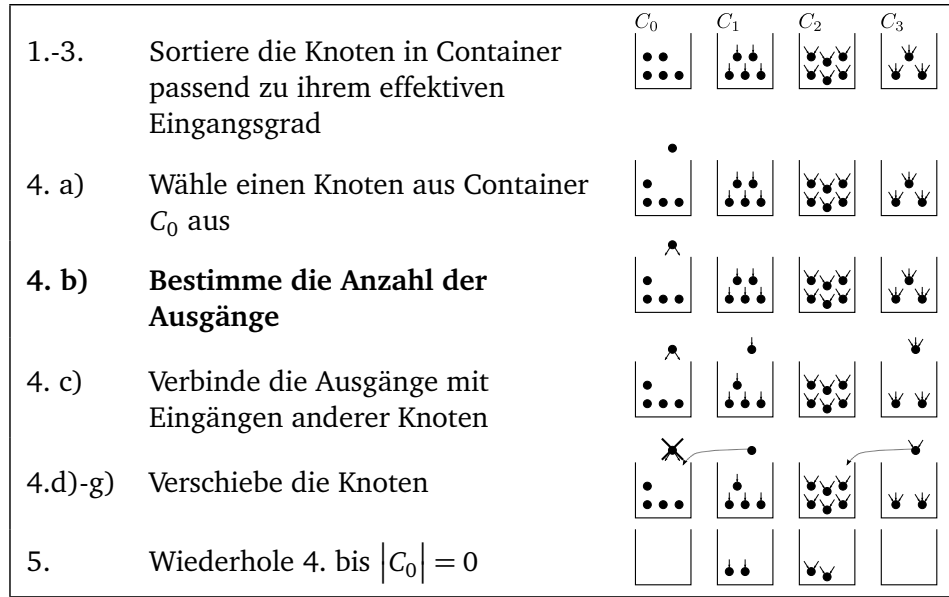
Das Verfahren lässt sich als stochastischer Prozess formulieren, bei dem alle Knoten basierend auf ihren Funktionen Containern  $C_i$  zugeordnet werden. Während der Algorithmus läuft, wird ein konkreter Graph aus dem Ensemble stückweise konstruiert, aber nur so weit festgelegt, wie es die Fragestellung verlangt. In Abbildung 3.1 ist sein Ablauf schematisch dargestellt. Änderungen der Grundform aus Abschnitt 1.2.3 sind hervorgehoben. Die neue Variante verfährt wie folgt:

1. Jedem der  $N$  Knoten des Netzwerkes wird eine Aktualisierungsfunktion, eine Anzahl von Eingängen **sowie eine Anzahl von Ausgängen** anhand der Charakteristik des zugrundeliegenden Ensembles zugeordnet. Um Korrelationen zu vermeiden, werden diese drei Zuordnungen unabhängig von einander getroffen.
2. Jeder Knoten wird anhand der Anzahl seiner Eingänge  $k_{\text{ein}} = i$  einem Container  $C_i$  zugeordnet. Ich werde die Anzahl von Knoten in Container  $C_i$  als  $|C_i|$  schreiben. Darüber hinaus bezeichne ich die Anzahl aller Knoten in den Containern als  $N_f = \sum_{i=0}^{i_{\text{max}}} |C_i|$  **und die Gesamtanzahl an irrelevanten Eingängen als  $k_{\text{ein}}^0$ , welche mit 0 initialisiert wird.** All diese Werte werden durch den Algorithmus verändert, im Speziellen wird  $N_f$  in jedem Schritt um 1 erniedrigt.
3. Entscheide für alle  $i > 0$  und für jeden Knoten in Container  $C_i$ , ob dieser eine konstante Funktion hat. Die Wahrscheinlichkeit einer konstanten Funktion mit  $i$  Eingängen bezeichne ich mit  $\beta(i)$ . Falls die Funktion

konstant ist, entferne diesen Knoten aus Container  $C_i$ , füge ihn in Container  $C_0$  hinzu **und erhöhe**  $k_{\text{ein}}^0$  **um**  $i$ .

4. Die folgenden Schritte werden wiederholt bis  $|C_0| = 0$  gilt:
  - a) Wähle einen Knoten  $A$  aus Container  $C_0$ .
  - b) **Bestimme die Anzahl der Ausgänge von  $A$  und bezeichne sie mit  $k_{\text{aus}}^{N_f}$ .**
  - c) **Wähle zufällig  $k_{\text{aus}}^{N_f}$  Eingänge aus allen  $\sum_i i |C_i| + k_{\text{ein}}^0$  noch vorhandenen Eingängen, um sie mit den Ausgängen von  $A$  zu verbinden.**
  - d) Falls  $m > 0$  der Eingänge eines Knotens  $B$  aus Container  $i \geq 1$  zum Knoten  $A$  verbunden wurden, bestimme die Wahrscheinlichkeit, dass die Aktualisierungsfunktion des Knotens  $B$  nun bei der reduzierten Anzahl Eingängen konstant ist. Diese Wahrscheinlichkeit wird im Fall  $m = 1$  abgekürzt mit  $\omega_i$ , im Allgemeinen ist sie mit  $1 - \sum_{k=i-m}^i (1 - \omega_k)$  gegeben, da sich diese Entscheidungen unabhängig nacheinander treffen lassen.
    - i. Falls die Funktion konstant ist, verschiebe  $B$  aus dem Container  $C_i$  in den Container  $C_0$  und erhöhe  $k_{\text{ein}}^0$  um  $i$ .
    - ii. Andernfalls verschiebe  $B$  aus dem Container  $C_i$  in den Container  $C_{i-m}$  und erhöhe  $k_{\text{ein}}^0$  um  $m$ .
  - e) **Senke  $k_{\text{ein}}^0$  um  $k_{\text{aus}}^{N_f}$ .**
  - f) **Insgesamt stellen die Modifikationen an  $k_{\text{ein}}^0$  sicher, dass die Gesamtanzahl an unverbundenen Eingängen  $\sum_{i \geq 1} i |C_i| + k_{\text{ein}}^0$  identisch zu der Anzahl der unverbundenen Ausgänge bleibt.**
  - g) Entferne den Knoten  $A$  aus dem System. Dies impliziert die Ersetzung  $N_f := N_f - 1$ .
5. Der finale Wert von  $N_f$  entspricht der Anzahl der Knoten, die nicht zum gefrorenen Kern des Netzwerkes gehören, dass durch diesen Algorithmus konstruiert wurde. Die Wahrscheinlichkeitsverteilung von  $N_f$  folgt dem stochastischen Prozess, der diesem Algorithmus entspricht.

Um die Konstruktion des Netzwerkes abzuschließen, müssten die übrig gebliebenen Ein- und Ausgänge ebenfalls zufällig verbunden werden, welche den nicht-gefrorenen Teil des Netzwerkes bilden. Dieses Vorgehen habe ich hier nicht weiter ausgeführt, da ich in der vorliegenden Arbeit den Schwerpunkt auf den gefrorenen Kern lege. In [12] wird auf diesen Aspekt weiter eingegangen, beispielsweise lassen sich die relevanten Knoten und das Skalenverhalten ihrer Anzahl ebenfalls ableiten.



**Abbildung 3.1.: Skizze des Containeralgorithmus**

Hier eine grobe Skizze des Containeralgorithmus aus Abschnitt 3.3.

### 3.4 Analytische Überlegungen

In diesem Abschnitt behandle ich vorerst nur das Funktionenensemble mit ausschließlich konstanten und invertierbaren Funktionen. In Abschnitt 3.5 werde ich begründen, warum die Argumente auch für andere Funktionenensembles gelten, insbesondere für die biased Funktionen.

Basierend auf dem Algorithmus aus Abschnitt 3.3 lässt sich die Anzahl der nicht-gefrorenen Knoten analytisch herleiten. Hierfür definiere ich den Parameter  $\epsilon = \frac{N_f}{N}$ , welcher den Anteil der bisher noch nicht gefrorenen Knoten widerspiegelt. Unter Vernachlässigung von Fluktuationen lässt sich die Anzahl der Knoten in Container  $i \geq 1$  wie in [24] ausdrücken als

$$|C_i| = \sum_{l=i}^{i_{\max}} |C_l^{\text{ini}}| \epsilon^i (1 - \epsilon)^{l-i} \binom{l}{i}, \quad (3.6)$$

wobei  $|C_l^{\text{ini}}|$  die Anzahl der Knoten in Container  $l$  am Ende vom 1. Schritt des Algorithmus ist. Wenn  $\epsilon$  klein wird, ist nur ein geringer Anteil der Eingänge noch vorhanden und die meisten Knoten, die sich zu Anfang in Container  $l$  befanden, befinden sich nun in Container  $i \ll l$ . Daher kann (3.6) für kleine  $\epsilon$  angenähert werden durch

$$|C_i| \simeq \epsilon^i \int_i^{i_{\max}} |C_l^{\text{ini}}| e^{-\epsilon l} l^i dl. \quad (3.7)$$

Durch die Kritikalitätsbedingung<sup>1</sup> (1.6) ist die initiale Anzahl von Knoten im Container  $C_0$  gegeben durch

$$|C_0^{\text{ini}}| = \sum_{i=1}^{i_{\max}} (i-1) |C_i^{\text{ini}}|, \quad (3.8)$$

was äquivalent zu der Bedingung

$$N_f = \sum_{i=0}^{i_{\max}} i |C_i| \quad (3.9)$$

ist. Diese ist in jedem Schritt des Algorithmus unter Vernachlässigung von Fluktuationen erfüllt. Im Mittel wird genau ein Eingang aus den Containern  $i \geq 1$  zum ausgewählten Knoten pro Schritt verbunden, da jeder der  $N_f K$  verbleibenden Eingänge zu den  $K$  Ausgängen des gewählten Knotens mit der gleichen Wahrscheinlichkeit verbunden wird. Daraus folgt, dass

$$|C_0| = \sum_{i=1}^{i_{\max}} (i-1) |C_i| \quad (3.10)$$

während jedes Schritts des Algorithmus gültig ist und die Größe aller Container gleichzeitig 0 wird, wenn auch  $\epsilon = 0$  wird.

Für kleine Werte von  $\epsilon$  ist, wie sich aus (3.6) ableiten lässt, die Summe in (3.10) durch den Summanden zu  $i = 2$  dominiert. Dies bedeutet, dass sich für kleine  $\epsilon$  fast alle nicht-gefrorenen Knoten in  $C_1$  oder  $C_2$  befinden und dies bis auf Fluktuationen

$$|C_0| \simeq |C_2| \quad (3.11)$$

gilt.

Durch zufällige Fluktuationen wird  $|C_0|$  typischerweise den Wert 0 erreichen, während  $N_f$  immer noch größer als 0 ist. Dies passiert, sobald die Standardabweichung von  $|C_0|$ , abgekürzt mit  $\sigma_{|C_0|}$ , in etwa dem Mittelwert von  $|C_0|$  entspricht. Dieser Mittelwert ist wiederum identisch zum Mittelwert von  $|C_2|$ . Dies ergibt für das Ende des Prozesses unter Berücksichtigung von Fluktuationen die Bedingung

$$|C_2| \simeq \sigma_{|C_0|}. \quad (3.12)$$

Um aus dieser Gleichung das Skalenverhalten von  $N_f$  mit  $N$  herzuleiten, muss ich beide Seiten in Abhängigkeit von  $N$  und  $N_f$  ausdrücken oder äquivalenter Weise in  $\epsilon$ .

<sup>1</sup> Diese Kritikalitätsbedingung für das Ensemble von konstanten und invertierbaren Funktionen muss hier für jeden einzelnen Grad individuell eingehalten werden, und nicht nur im Mittel über alle Knoten.

Im Folgenden werde ich die Varianz  $\sigma_{|C_0|}^2$  der Anzahl der Knoten in Container  $C_0$  herleiten. In jedem Zeitschritt wird genau ein Knoten aus dem System entfernt und einige der Knoten aus den Containern  $C_i$  mit  $i > 0$  nach  $C_0$  verschoben. Die Gesamtanzahl der Knoten in allen Containern  $N_f$  verringert sich bei jeder Iteration des Algorithmus exakt um 1. Fluktuationen  $\sigma_{N_{\text{rf}}}$  der Gesamtanzahl  $N_{\text{rf}} := N_f - |C_0|$  der Knoten in  $C_i$  mit  $i > 0$  sind Abweichungen vom Mittelwert, der in (3.7) gegeben ist. Diese Abweichungen sind, bis auf das Vorzeichen, identisch mit den Fluktuationen  $\sigma_{|C_0|}$  in der Anzahl der Knoten in Container  $C_0$ , da die Gesamtanzahl der Knoten in allen Containern  $N_f$  deterministisch abnimmt und daher keine zufälligen Fluktuationen zeigt. Da sich für kleine  $\epsilon$  die überwiegende Mehrheit der übrigen Knoten in  $C_1$  befindet, wird die Fluktuation  $\sigma_{N_{\text{rf}}}$  in der Anzahl der Knoten in den Containern  $C_i$  mit  $i > 0$  durch Fluktuationen  $\sigma_{|C_1|}$  im Container  $C_1$  dominiert. Dies führt zu

$$\sigma_{|C_0|} = \sigma_{N_{\text{rf}}} \simeq \sigma_{|C_1|}. \quad (3.13)$$

Es gibt zwei Beiträge zu dieser Varianz:

1. Die Anzahl der Knoten  $N_{\text{rf}}$ , die in den Containern  $C_i$  mit  $i \geq 1$  verbleiben, muss im Mittel der Anzahl der verbleibenden Ausgänge geteilt durch deren mittleren Grad entsprechen. Dies gibt folgenden Beitrag zur Varianz:

$$\sigma_{N_{\text{rf}}}^2 \simeq \frac{N_{\text{rf}}}{K} \sigma_{k_{\text{aus}}}^2 \propto N_{\text{rf}} \sigma_{k_{\text{aus}}}^2. \quad (3.14)$$

2. Der zweite Beitrag entspringt der Tatsache, dass die Anzahl der nicht-gefrorenen Eingänge für kleine  $\epsilon$  poissonverteilt ist. Die Varianz dieser Verteilung ist identisch zu ihrem Mittelwert, welcher wiederum proportional zu  $N_{\text{rf}}$  ist. Da der größte Teil der Knoten nur einen Eingang hat, ist  $N_{\text{rf}}$  ebenfalls die Varianz von  $|C_1|$ , was zusammen mit (3.13) zu  $\sigma_{N_{\text{rf}}}^2 \propto N_{\text{rf}}$  führt.

Für Netzwerke mit einer skalenfreien Ausgangsgradverteilung und einem Exponenten zwischen 2 und 3 dominiert der erste Term. Er ist proportional zu  $N_{\text{rf}}$  multipliziert mit einer Potenz von  $N$ . Andernfalls ergeben der erste und zweite Term zusammen  $\sigma_{N_{\text{rf}}}^2 \propto N_{\text{rf}}$ . Ich folgere

$$\sigma_{|C_0|} \propto \sqrt{N_{\text{rf}}} \sigma_{k_{\text{aus}}} \propto \sqrt{N_{\text{rf}}} N^A \quad (3.15)$$

mit einem Exponenten  $A$ , der von der Ausgangsgradverteilung abhängt. Wenn die Varianz der Ausgangsgradverteilung endlich ist, erhalte ich  $A = 0$ . Dieser Fall tritt unter anderem ein, wenn der Ausgangsgrad aller Knoten identisch, poisson- oder nach einem Potenzgesetz mit einem Exponenten

$\gamma_{\text{aus}} \geq 3$  verteilt ist. Falls die Ausgangsgradverteilung ein Potenzgesetz mit  $\gamma_{\text{aus}} \in (2, 3)$  ist, gilt

$$\sigma_{k_{\text{aus}}} = \sum_{k=0}^{k_{\text{aus}}^{\text{max}}} P_{\text{aus}}(k) (k - K)^2 \propto (k_{\text{aus}}^{\text{max}})^{3-\gamma_{\text{aus}}}, \quad (3.16)$$

was zusammen mit (3.4) oder (3.5) zu  $A = 3 - \gamma_{\text{aus}}$  oder  $A = (3 - \gamma_{\text{aus}})/\gamma_{\text{aus}}$  führt. Für  $\gamma_{\text{aus}} = 3$  fallen diese drei Gleichungen zusammen und ergeben konsistent  $A = 0$ .

Um die Rechnung abschließen zu können, fehlt noch ein Ausdruck für  $|C_2|$  in (3.12). Dieser Ausdruck ist in (3.7) enthalten und wurde bereits in [24] für den Fall der Poisson-Ausgangsgradverteilungen angegeben. Wenn die Eingangsgradverteilung ein endliches zweites Moment hat wie im Fall von konstanter, Poisson- oder skalenfreier Eingangsgradverteilungen mit  $\gamma_{\text{ein}} \geq 3$ , konvergiert das Integral in (3.7), selbst wenn ich den Cut-off auf unendlich und  $\epsilon$  auf 0 setze. Dies führt zu

$$|C_2| \propto N \epsilon^2. \quad (3.17)$$

Für Netzwerke mit skalenfreier Eingangsgradverteilung können wir (3.7) umschreiben in

$$|C_i| \propto \epsilon^i N \int_i^{i_{\text{max}}} e^{-\epsilon l} l^{i-\gamma_{\text{ein}}} dl. \quad (3.18)$$

Im Falle von  $\gamma_{\text{ein}} \geq 3$  konvergiert das Integral, was wieder zu (3.17) führt. Im Falle von  $\gamma_{\text{ein}} \in (2, 3)$  divergiert das zweite Moment, und womit der Wert des Integrals entweder durch die obere Schranke des Integrals  $k_{\text{ein}}^{\text{max}}$  oder durch die inverse exponentielle Abklingkonstante  $\frac{1}{\epsilon}$  bestimmt ist, je nachdem, was kleiner ist. Zusammen ergibt dies

$$|C_2| \propto N \epsilon^2 \left( \min \left( k_{\text{ein}}^{\text{max}}, \frac{1}{\epsilon} \right) \right)^{3-\gamma_{\text{ein}}}. \quad (3.19)$$

Im Falle von  $k_{\text{ein}}^{\text{max}} > \frac{1}{\epsilon}$  erhalte ich

$$|C_2| \propto N \epsilon^{\gamma_{\text{ein}}-1}, \quad (3.20)$$

unabhängig davon, wie  $k_{\text{ein}}^{\text{max}}$  mit  $N$  skaliert. Für  $k_{\text{ein}}^{\text{max}} < \frac{1}{\epsilon}$  erhalte ich für  $k_{\text{ein}}^{\text{max}} \propto N$

$$|C_2| \propto N^{4-\gamma_{\text{ein}}} \epsilon^2 \quad (3.21)$$

und im Falle von  $k_{\text{ein}}^{\text{max}} \propto N^{\frac{1}{\gamma_{\text{ein}}}}$

$$|C_2| \propto N^{\frac{3}{\gamma_{\text{ein}}}} \epsilon^2. \quad (3.22)$$

Für  $\gamma_{\text{ein}} = 3$  ergeben alle vier Ausdrücke für  $|C_2|$  das gleiche Ergebnis, wie es konsistenter Weise erwartet wurde.

Diese vier Fälle können zusammengefasst werden zu

$$|C_2| \propto N^a \epsilon^b \quad (3.23)$$

mit unterschiedlichen Termen für  $a$  und  $b$ .

Wenn ich dieses Ergebnis zusammen mit (3.15) in (3.12) einsetze, erhalte ich das gesuchte Skalengesetz für die Anzahl der nicht-gefrorenen Knoten  $N_{\text{nf}}$ , die identisch mit dem finalen Wert von  $N_f$  bzw.  $N_{\text{ff}}$  sind, als

$$N_{\text{nf}} \propto \epsilon N \propto N^{1 - \frac{\frac{1}{2} + A - a}{\frac{1}{2} - b}} =: N^x. \quad (3.24)$$

Da sowohl  $k_{\text{ein}}^{\text{max}}$  als auch  $\frac{1}{\epsilon}$  auf eine nichttriviale Art und Weise mit  $N$  skalieren und auch das Skalieren von  $\epsilon$  mit  $N$  von dem Ergebnis von (3.19) abhängt, ist es nicht immer möglich, im Voraus zu bestimmen, ob  $k_{\text{ein}}^{\text{max}}$  größer oder kleiner ist als  $\frac{1}{\epsilon}$ . In solchen Fällen nehme ich zuerst eine Version der Ungleichung an. Falls dies zu einer Inkonsistenz führt, nutze ich die andere Version. Falls  $k_{\text{ein}}^{\text{max}} \propto N^z$  ist und ich die Tatsache nutze, dass  $\epsilon \propto N^{x-1}$  (mit  $x$  definiert in (3.24)), lässt sich die Konsistenzbedingung schreiben als

$$k_{\text{ein}}^{\text{max}} \lesseqgtr \frac{1}{\epsilon} \Leftrightarrow z + x - 1 \lesseqgtr 0. \quad (3.25)$$

Im Standard-Fall mit endlichem zweiten Moment der Ein- und Ausgangsgradverteilung habe ich  $A = 0$ ,  $a = 1$ , und  $b = 2$ . Dies führt zu dem bereits bekannten Ergebnis

$$N_{\text{nf}} \propto N^{\frac{2}{3}}. \quad (3.26)$$

Auf eine ähnliche Art und Weise lassen sich Skalengesetze für  $|C_2|$  finden. Hierfür muss das Skalenverhalten von  $\epsilon$  mit  $N$  aus (3.24) mit (3.23) kombiniert werden. Hieraus erhält man

$$|C_2| \propto N^{a+b \frac{\frac{1}{2} + A - a}{b - \frac{1}{2}}} =: N^y. \quad (3.27)$$

Da ich drei Fälle für jeweils Ein- und Ausgangsgradverteilungen betrachte, habe ich insgesamt 9 verschiedene Skalenverhalten für die Anzahl der nicht-gefrorenen Knoten mit der Gesamtanzahl der Knoten in einem kritischen Netzwerk  $N^x$ . In manchen Fällen sind, wie diskutiert, Fallunterscheidungen notwendig, weshalb einige dieser 9 Fälle sogar noch weiter differenziert werden müssen. Ebenfalls habe ich für alle diese Fälle einen Ausdruck für den Skalenexponenten  $y$  hergeleitet, der den Anteil der nicht-gefrorenen Knoten mit zwei nicht-gefrorenen Eingängen beschreibt.



Die Ausdrücke für  $x$  und  $y$  in den unterschiedlichen Fällen sind in Tabelle 3.1 aufgeführt und in Abbildung 3.2 grafisch dargestellt. In Abbildung 3.2 wird darüber hinaus auch  $\frac{y}{x}$  und das Kriterium  $k_{\text{ein}}^{\max} \leq \frac{1}{\epsilon}$  gezeigt. Die markanteste Eigenschaft dieser Ergebnisse ist, dass der Skalenexponent mit sinkendem  $\gamma_{\text{ein}}$  anwächst, aber mit sinkendem  $\gamma_{\text{aus}}$  ebenfalls sinkt. Darüber hinaus wächst der Skalenexponent  $y$ , der den Anteil der nicht-gefrorenen Knoten mit zwei nicht-gefrorenen Eingängen beschreibt, ebenfalls mit sinkendem  $\gamma_{\text{aus}}$  an.

Folgende Möglichkeiten, diese Ergebnisse auf ihre Korrektheit zu überprüfen, wurden ausgeschöpft:

- Jeder Ausdruck für  $x$  muss den Wert  $\frac{2}{3}$  ergeben wenn  $\gamma_{\text{ein}} = \gamma_{\text{aus}} = 3$ . Falls sowohl Ein- als auch Ausgangsgradverteilung skalenfrei sind, muss  $x$  den Wert der zugehörigen Poissonverteilung annehmen, wenn nur eines der beiden  $\gamma$  auf 3 gesetzt wird.
- $x$  muss für  $\gamma \in [2, 3]$  im Intervall  $[0, 1]$  liegen.
- Für skalenfreie Eingangsgradverteilungen muss mindestens eine der zwei möglichen  $x$ -Werte die Konsistenzbedingung in (3.25) erfüllen. Falls beide diese Bedingung erfüllen, müssen sie identisch sein.
- Für den Grenzfall aus (3.25) folgt  $x = 1 - z$ . Hieraus lassen sich zwei weitere Korrektheitskriterien ableiten:
  - Aus  $z = 1$ , also  $k_{\text{ein}}^{\max} \propto N$ , ergibt sich, dass  $x = 0$  nur auf dem Rand des betrachteten Bereichs  $\gamma_{\text{ein}}, \gamma_{\text{aus}} \in [2, 3]$  möglich ist, da dies ansonsten ungültige  $x < 0$  zu Folge hätte.
  - Für  $z = \frac{1}{\gamma_{\text{ein}}}$  muss  $x$  für den Grenzfall aus (3.25) den Wert  $1 - \frac{1}{\gamma_{\text{ein}}}$  unabhängig von der Ausgangsgradverteilung annehmen.

Um auch die Ergebnisse für  $y$  auf ihre Korrektheit hin zu überprüfen, kann ich analog vorgehen, mit dem Unterschied, dass  $y$  den Wert  $\frac{1}{3}$  für  $\gamma = 3$  annehmen muss. Als finalen Test habe ich Simulationen, wie in Abschnitt 3.6 beschrieben, durchgeführt.

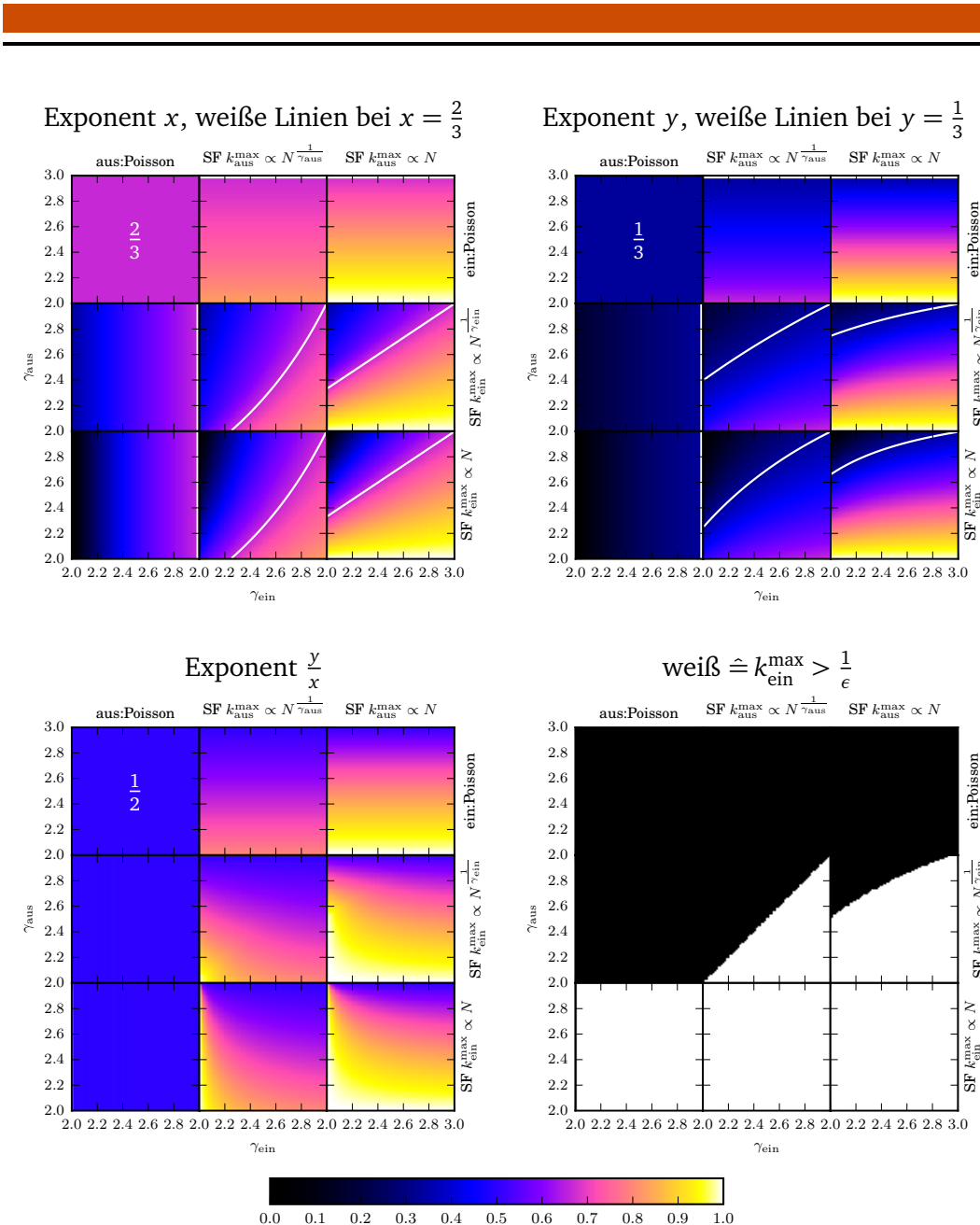
### 3.5 Erweiterung auf andere Ensembles von Aktualisierungsfunktionen

Alle bisherigen analytischen Ergebnisse in diesem Kapitel sind mit einem Funktionenensemble aus konstanten und invertierbaren Funktionen erzielt worden, obwohl der vorgestellte Algorithmus solche Einschränkungen nicht hat. Die analytischen Überlegungen müssten für andere Funktionenensembles leicht angepasst werden, würden aber aus folgenden Gründen zum gleichen Ergebnis führen: Gleichung (3.6) besagt immer noch, dass gegen Ende des Algorithmus nur noch die Container  $C_0$ ,  $C_1$  und  $C_2$  berücksichtigt werden müssen. Knoten in  $C_0$  oder  $C_1$  verhalten sich identisch zu denen im Falle konstanter oder invertierbarer Aktualisierungsfunktionen. Der einzige Unterschied ist, dass ein Anteil  $\omega_2$  von Knoten aus  $C_2$  bereits komplett einfriert,

$x$	Aus $\rightarrow$	Poisson	SF $k_{\text{aus}}^{\text{max}} \propto N^{\frac{1}{2\gamma_{\text{aus}}}}$	SF $k_{\text{aus}}^{\text{max}} \propto N^{\frac{3-\gamma_{\text{aus}}}{2}}$
Ein $\downarrow$		$\sigma _{k_{\text{aus}}} \propto 1$	$\sigma _{k_{\text{aus}}} \propto N^{\frac{3-\gamma_{\text{aus}}}{2\gamma_{\text{aus}}}}$	$\sigma _{k_{\text{aus}}} \propto N^{\frac{3-\gamma_{\text{aus}}}{2}}$
Poisson	$ C_2  \propto N\epsilon^2$	$\frac{2}{3}$	$\frac{1}{3} + \frac{1}{\gamma_{\text{aus}}}$	$\frac{5-\gamma_{\text{aus}}}{3}$
SF $k_{\text{ein}}^{\text{max}} \propto N^{\frac{1}{\gamma_{\text{ein}}}}$	$ C_2  \propto N^{\frac{3}{\gamma_{\text{ein}}}} \epsilon^2$	$\frac{4}{3} - \frac{2}{\gamma_{\text{ein}}}$	$1 + \frac{1}{\gamma_{\text{aus}}} - \frac{2}{\gamma_{\text{ein}}}$	$\frac{7-\gamma_{\text{aus}}}{3} - \frac{2}{\gamma_{\text{ein}}}$
	$x \leq 1 - \frac{1}{\gamma_{\text{ein}}}$		$\gamma_{\text{ein}} \leq \gamma_{\text{aus}}$	$\gamma_{\text{ein}} \leq \frac{3}{4-\gamma_{\text{aus}}}$
	$x \geq 1 - \frac{1}{\gamma_{\text{ein}}}$		$\gamma_{\text{ein}} = \gamma_{\text{aus}}$	$\gamma_{\text{ein}} = \frac{3}{4-\gamma_{\text{aus}}}$
SF $k_{\text{ein}}^{\text{max}} \propto N$	$ C_2  \propto N\epsilon^{\gamma_{\text{ein}}-1}$		$1 - \frac{1-\frac{3-\gamma_{\text{aus}}}{\gamma_{\text{aus}}}}{2\gamma_{\text{ein}}-3}$	$1 - \frac{2-\gamma_{\text{aus}}}{3-2\gamma_{\text{ein}}}$
	$x \geq 1 - \frac{1}{\gamma_{\text{ein}}}$		$\gamma_{\text{ein}} \geq \gamma_{\text{aus}}$	$\gamma_{\text{ein}} \geq \frac{3}{4-\gamma_{\text{aus}}}$
		$\frac{2\gamma_{\text{ein}}-4}{2\gamma_{\text{ein}}-3}$	immer	immer
	$ C_2  \propto N\epsilon^{\gamma_{\text{ein}}-1}$		$1 - \frac{2-\gamma_{\text{aus}}}{3-2\gamma_{\text{ein}}}$	$1 - \frac{2-\gamma_{\text{aus}}}{3-2\gamma_{\text{ein}}}$
		$\frac{2\gamma_{\text{ein}}-4}{2\gamma_{\text{ein}}-3}$	$\gamma_{\text{ein}} = \gamma_{\text{aus}}$	$\gamma_{\text{ein}} = \gamma_{\text{aus}}$
$y$	Aus $\rightarrow$	Poisson	SF $k_{\text{aus}}^{\text{max}} \propto N^{\frac{1}{2\gamma_{\text{aus}}}}$	SF $k_{\text{aus}}^{\text{max}} \propto N^{\frac{3-\gamma_{\text{aus}}}{2}}$
Ein $\downarrow$		$\sigma _{k_{\text{aus}}} \propto 1$	$\sigma _{k_{\text{aus}}} \propto N^{\frac{3-\gamma_{\text{aus}}}{2\gamma_{\text{aus}}}}$	$\sigma _{k_{\text{aus}}} \propto N^{\frac{3-\gamma_{\text{aus}}}{2}}$
Poisson	$ C_2  \propto N\epsilon^2$	$\frac{1}{3}$	$\frac{2}{\gamma_{\text{aus}}} - \frac{1}{3}$	$\frac{7-2\gamma_{\text{aus}}}{3}$
SF $k_{\text{ein}}^{\text{max}} \propto N^{\frac{1}{\gamma_{\text{ein}}}}$	$ C_2  \propto N^{\frac{3}{\gamma_{\text{ein}}}} \epsilon^2$	$\frac{2}{3} - \frac{1}{\gamma_{\text{ein}}}$	$\frac{2}{\gamma_{\text{aus}}} - \frac{1}{\gamma_{\text{ein}}}$	$\frac{8-2\gamma_{\text{aus}}}{3} - \frac{1}{\gamma_{\text{ein}}}$
	$x \leq 1 - \frac{1}{\gamma_{\text{ein}}}$		$\gamma_{\text{ein}} \leq \gamma_{\text{aus}}$	$\gamma_{\text{ein}} \leq \frac{3}{4-\gamma_{\text{aus}}}$
	$x \geq 1 - \frac{1}{\gamma_{\text{ein}}}$		$\gamma_{\text{ein}} = \gamma_{\text{aus}}$	$\gamma_{\text{ein}} = \frac{3}{4-\gamma_{\text{aus}}}$
SF $k_{\text{ein}}^{\text{max}} \propto N$	$ C_2  \propto N\epsilon^{\gamma_{\text{ein}}-1}$		$1 - \frac{1-\frac{\gamma_{\text{aus}}(3-2\gamma_{\text{ein}})}{\gamma_{\text{aus}}(3-2\gamma_{\text{ein}})}}{\gamma_{\text{ein}}}$	$1 - \frac{1-\frac{\gamma_{\text{aus}}(3-2\gamma_{\text{ein}})}{\gamma_{\text{aus}}(3-2\gamma_{\text{ein}})}}{\gamma_{\text{ein}}}$
	$x \geq 1 - \frac{1}{\gamma_{\text{ein}}}$		$\gamma_{\text{ein}} \geq \gamma_{\text{aus}}$	$\gamma_{\text{ein}} \geq \frac{3}{4-\gamma_{\text{aus}}}$
		$\frac{\gamma_{\text{ein}}-2}{2\gamma_{\text{ein}}-3}$	immer	immer
	$ C_2  \propto N\epsilon^{\gamma_{\text{ein}}-1}$		$1 - \frac{1-\frac{\gamma_{\text{aus}}(3-2\gamma_{\text{ein}})}{\gamma_{\text{aus}}(3-2\gamma_{\text{ein}})}}{\gamma_{\text{ein}}}$	$1 - \frac{1-\frac{\gamma_{\text{aus}}(3-2\gamma_{\text{ein}})}{\gamma_{\text{aus}}(3-2\gamma_{\text{ein}})}}{\gamma_{\text{ein}}}$
		$\frac{\gamma_{\text{ein}}-2}{2\gamma_{\text{ein}}-3}$	$\gamma_{\text{ein}} = \gamma_{\text{aus}}$	$\gamma_{\text{ein}} = \gamma_{\text{aus}}$

**Tabelle 3.1.: Tabelle der Exponenten für skalenfreie Netze**

Je 9 verschiedene Fälle für die Skalensexponenten  $x$  und  $y$ , die das Skalenverhalten  $N_{\text{nf}} \propto N^x$  der Anzahl der nicht-gefrorenen Knoten  $N_{\text{nf}}$  mit  $N$  und das Skalenverhalten der Anzahl der nicht-gefrorenen Knoten mit zwei nicht-gefrorenen Eingängen  $|C_2| \propto N^y$  mit  $N$  widerspiegeln. Die Ausdrücke für die Spezialfälle  $\gamma_{\text{ein}} = \gamma_{\text{aus}}$  sind ebenfalls meistens angegeben. Immer wenn es nötig war, sind zwei Ausdrücke für  $x$  bzw.  $y$  zusammen mit ihren Gültigkeitsbedingungen gegeben, andere Fälle sind durch das Wort *immer* gekennzeichnet. Auch sind für den Grenzfall zwischen den Bedingungen die Exponenten noch einmal explizit angegeben. Alle Exponenten sind in Abbildung 3.2 visualisiert.



**Abbildung 3.2.: Grafische Darstellung der 9 Fälle aus Tabelle 3.1**

Jede der 4 Grafiken besteht aus 3x3 Unterabbildungen. Hier sind jeweils für Ein- und Ausgangsgradverteilung die Fälle Poisson, skalenfrei (SF) mit Cut-off  $k^{\text{max}} \propto N^{\frac{1}{3}}$  und skalenfrei (SF) mit Cut-off  $k^{\text{max}} \propto N$  unterschieden. Für die skalenfreien Eingangsgradverteilungen zeigt die x-Achse einen Exponenten  $\gamma_{\text{ein}}$  und für die skalenfreien Ausgangsgradverteilungen die y-Achse den Exponenten  $\gamma_{\text{aus}}$ . Eine Legende für den Farbwert ist ganz unten abgebildet.

In der ersten Abbildung wird der Exponent  $x$  zugehörig zu  $N_{\text{nf}} \propto N^x$  gezeigt. Weiße Linien geben den klassischen Exponenten  $\frac{2}{3}$  an.

In der zweiten Abbildung wird der Exponent  $y$  zugehörig zu  $|C_2| \propto N^y$  gezeigt. Weiße Linien geben den klassischen Exponenten  $\frac{1}{3}$  an.

In der dritten Abbildung wird das Verhältnis  $\frac{y}{x}$  dargestellt. Klassische Netzwerke haben einen Wert von  $\frac{1}{2}$  (blau).

In der vierten Abbildung wird schließlich gezeigt, welcher Wert das Minimum aus (3.19) bestimmt: weiß  $\hat{=} k_{\text{ein}}^{\text{max}} > \frac{1}{\epsilon}$ , schwarz  $\hat{=} k_{\text{ein}}^{\text{max}} \leq \frac{1}{\epsilon}$ .

selbst wenn nur einer seiner Eingänge mit einem gefrorenen Knoten verbunden wird. Solche Knoten könnten Prinzipiell auch in Container  $C_1$  platziert werden. Die Simulation wäre dann identisch zu derjenigen, die nur konstante und invertierbare Funktionen betrachtet. Da  $|C_1| \gg |C_2|$  ist, macht dies keinen signifikanten Unterschied in den Berechnungen und hat somit auch keine Auswirkungen auf die gefundenen Skalengesetze.

### 3.6 Numerische Simulationen

Um meine analytischen Berechnungen zu bestätigen, habe ich den beschriebenen Algorithmus implementiert und mit ihm das Einfrieren simuliert. Anstelle von Poissonverteilungen verwendete ich einen festen Wert für die Gradverteilung. Dies macht keinen Unterschied, da beide Verteilungen ein nicht divergierendes zweites Moment besitzen. Wenn der für  $K$  benötigte Wert keine ganze Zahl war, wurde eine Mischung aus den zwei benachbarten ganzzahligen Graden als Verteilung für  $k$  genommen. Es stellte sich im Allgemeinen als schwierig heraus, zwei Verteilungen zu erzeugen, die neben vielen offensichtlichen Parametern auch noch zusätzlich in der Gesamtanzahl der Kanten exakt übereinstimmen. Falls nur eine der Verteilungen skalenfrei war, konnte die jeweils andere leicht mit passender Kantenanzahl erzeugt werden. Im Falle dass Ein- und Ausgangsgradverteilung vom identischen Typ und skalenfrei waren, habe ich diese nur mit exakt gleichem Exponenten untersucht und, bis auf Permutation, ein und dieselbe Stichprobe der Verteilung der Ein- und Ausgangsgrade genommen.

In Abschnitt 3.2 erwähnte ich, dass das Erstellen einer Stichprobe, deren Cut-off wie  $k^{\max} \propto N^{\frac{1}{\gamma}}$  skaliert, nicht trivial mit korrektem Stichprobenumfang  $N$  und korrektem Mittelwert  $K$  möglich ist. Die beiden Freiheitsgrade  $c$  und  $P(k=2)$  ließen sich allerdings mit einem iterativen Verfahren sehr effizient bestimmen. Dieses terminiert, nachdem  $N$  bis auf einen relativen Fehler von  $10^{-4}$  und  $K$  bis auf einen relativen Fehler von  $10^{-3}$  die vorgegebenen Werte erreicht haben.

Es stellte sich als schwierig heraus, jegliche Korrelationen zwischen Ein- und Ausgangsgrad sowie der Booleschen Aktualisierungsfunktion zu vermeiden. Daher musste bei vielen Details auf eine korrekte Implementierung der Stochastik geachtet werden, weil sonst leicht Korrelationen oder Vorzugsergebnisse entstanden wären. Die Simulation wurde wie folgt durchgeführt. Für jeden Satz an Ein- und Ausgangsgradverteilungen und jeden betrachteten Wert von  $\gamma$  habe ich den Algorithmus zwischen  $10^3$  und  $10^6$  mal ausgeführt für Werte  $N \in \{2^{10}, 2^{11}, \dots, 2^{20}, \dots\}$ . Hieraus erhielt ich eine Verteilung der Anzahl der nicht-gefrorenen Knoten  $N_{\text{nf}}$  und der finalen Anzahl der Knoten in Container  $C_2$ . Die Daten wurden mittels logarithmischen Binning mit einer Schrittweite von 1,05 geglättet. Im linken Teil der Kurven, die in log-log Darstellung nur eine geringe und annähernd konstante Steigung haben,

<sup>2</sup> Der Algorithmus wurde für große Werte von  $N$  aufgrund langer Rechenzeiten seltener ausgeführt.

wurden benachbarte Bins soweit zusammengefasst, dass sie alle in etwa die gleiche Anzahl an Ereignissen enthielten wie das häufigste Bin. Hierdurch hat sich eine weitere Glättung ergeben, ohne auf eine Aussagekraft verzichten zu müssen.

Ein typisches Ergebnis visualisiert Abbildung 3.3, wobei hier unterschiedliche Exponenten für die Skalierung der Achsen gezeigt sind. Die universelle Kurvenform wurde in [12] hergeleitet und spielt in dieser Arbeit keine weitere Rolle. Die Form der Kurve ist von der jeweiligen Eingangs- und Ausgangsgradverteilung abhängig, nicht aber von  $N$ . Falls sich die Form doch mit  $N$  ändert, ist  $N$  noch nicht groß genug gewählt, so dass weiterhin finite-size Effekte eine Rolle spielen. Alternativ kann dies auch aus Problemen der Containermethode mit zu großem mittlerem Grad  $K$  herrühren, die in Kapitel 2 untersucht wurden.

Die Qualität des Datenkollapses ist allerdings nicht immer so gut, insbesondere wenn  $\gamma$  sich 2 annähert oder wenn  $N$  klein ist (finite-size Effekte). Um den gefundenen analytischen Ausdruck, der im Grenzfall  $N \rightarrow \infty$  exakt sein sollte, mit meinen Daten zu vergleichen, habe ich ein automatisches Verfahren entwickelt, um die Qualität eines Datenkollapses zu quantifizieren. Dieses Verfahren nimmt zwei Kurven  $N_1 = 2^n$  und  $N_2 = 2^{n+1}$  und testet gerastert mögliche Kollapsexponenten  $w \in (0, 1)$ . Dieses automatische Verfahren liefert einen optimalen ( $N$ -abhängigen) Wert für den Exponenten, der sich dem theoretischen Wert mit wachsendem  $N$  annähern sollte. Die Qualität eines Kollapses wird mit einem Fitnesswert  $u(w)$  quantifiziert, welcher einer Approximation der absoluten Fläche zwischen den zwei interpolierten Kurven entspricht, wobei die ersten 20 % der Kurven (typischerweise sehr flach und wenig aussagekräftig) ausgelassen wurden. Analog lässt sich auch für mehr als zwei Kurven verfahren, indem die Fläche zwischen der jeweils äußeren Begrenzung der Kurvenschar genommen wird (siehe Abbildung 3.3). Zahlreiche Beispiele suggerieren, dass ein höherer Fitnesswert auch dem entspricht, was man intuitiv als besseren Kollaps bezeichnen würde.

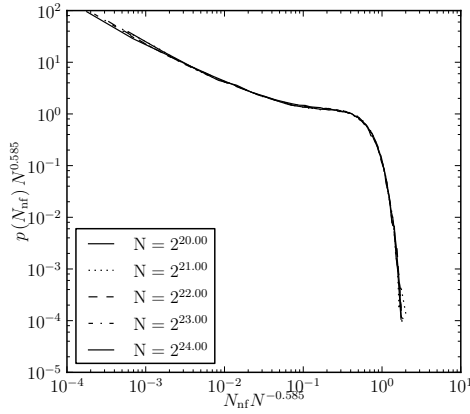
Um zu visualisieren, wie gut ein Skalenexponent zwei Kurven zum Kollaps bringt, wird  $u(w)$  mittels Farben kodiert, siehe dazu Abbildung 3.4. Die Farben wurden so gewählt, dass der Verlauf von Blau über Gelb nach Weiß dem Intervall  $[0, 1]$  entspricht und

$$\tilde{u}(w) := \min \left( 1, 2 \sqrt{\frac{u(w) - u^{\min}}{u^{\max} - u^{\min}}} \right), \quad (3.28)$$

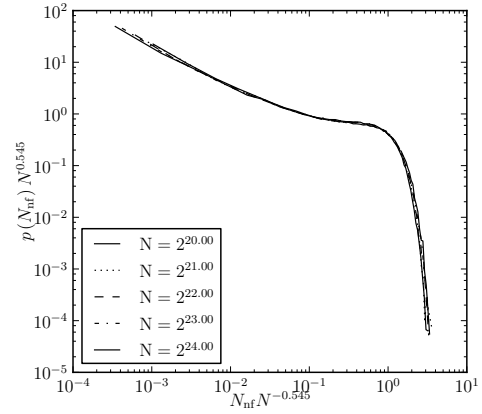
zugeordnet. Diese Transformation des Fitnesswertes  $u \mapsto \tilde{u}$  hat sich heuristisch unter Anderem deshalb als gut herausgestellt, weil sie nahe des Minimums  $u(w)$  eine hohe Farbauflösung liefert.

Die Ergebnisse des Vergleichs zwischen analytischen und numerischen Daten sind in Abbildung 3.5 mit diesem Farbschema visualisiert. Jede der Unterabbildungen zeigt auf der y-Achse einen möglichen Skalenexponenten, der für einen Kollaps genutzt werden könnte, und auf der x-Achse den Wert von

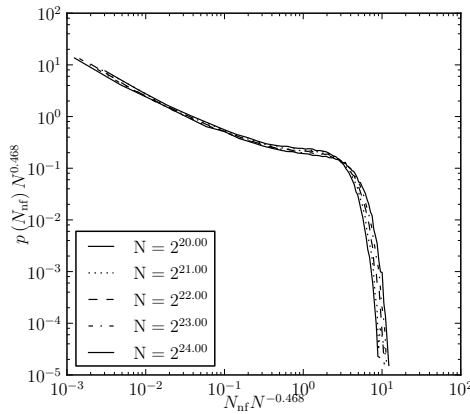
numerisch optimaler Kollaps  $x = 0.585$



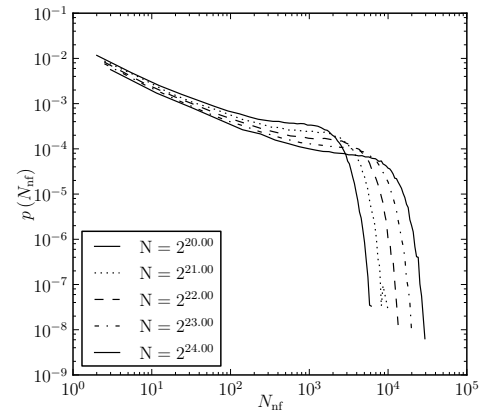
theoretischer Exponent  $x = 0.545$



Bsp. schlechter Kollaps  $x = 0.468$



unskalierte Daten

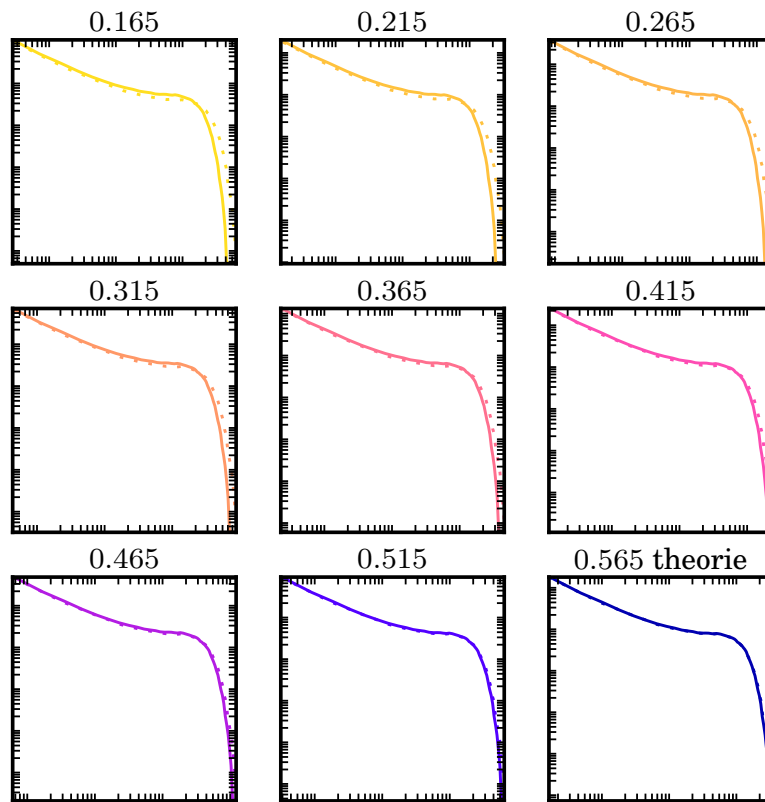


**Abbildung 3.3.: Typisches Beispiel für die Wahrscheinlichkeitsverteilung für die Anzahl der nicht-gefrorenen Knoten**

Diese Daten stammen von einem kritischen RBN mit skalenfreier Ein- und Ausgangsgradverteilung mit  $\gamma = 2.2$  und einem Cut-off, der wie  $k^{\max} \propto N^{\frac{1}{\gamma}}$  skaliert. Gemittelt wurde jeweils über  $\approx 10^5$  Realisierungen. Der numerisch optimale Exponent wurde hier nicht paarweise für zwei Kurven bestimmt, wie an allen anderen Stellen in dieser Arbeit, sondern simultan für alle Kurven in der Abbildung.

$N_1$  des untersuchten Kurvenpaares. Einige Unterabbildungen enthalten weiße Bereiche aufgrund fehlender Simulationsdaten oder aufgrund zu starker Fluktuation wegen einer ungenügenden Statistik. Der theoretische Exponent liegt in fast allen Unterabbildungen nahe am numerisch optimalen Skalensexponenten  $\arg \min_w u(w)$ . In einigen Fällen kann man erkennen, dass sich dieser numerisch optimale Bereich mit wachsender Netzwerkgröße  $N_1$  nach oben oder unten verschiebt, was finite-size Effekten entspricht. In diesen Fällen kann man erkennen, dass sich der optimal Kollapsbereich jedoch den analytischen Ergebnissen annähert und diese somit für den Grenzfall  $N \rightarrow \infty$  noch plausibler macht.

Diese Auswertung habe ich ebenfalls für den Skalensexponenten  $y$  von  $|C_2|$  durchgeführt. In Abbildung 3.6 werden die Ergebnisse dargestellt. Auch hier stimmen Theorie und Numerik sehr gut überein.

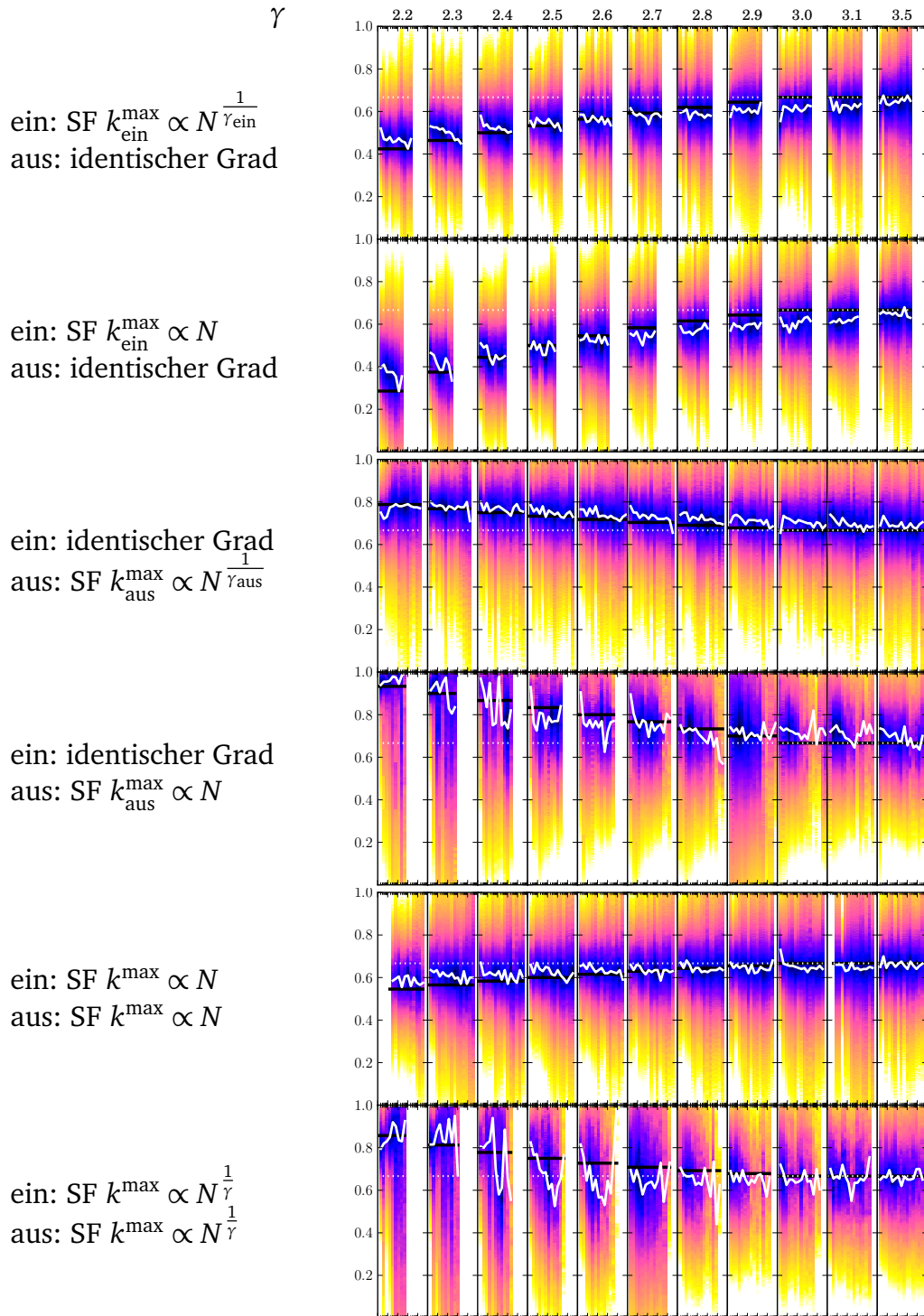


**Abbildung 3.4.: Legende für die Farbkodierung in Abbildung 3.5 und 3.6 anhand von Beispielen**

Um zu visualisieren, wie gut ein Skalensexponent zwei Kurven zum Kollaps bringt, wird die eingeführte Metrik  $u(w)$  mittels Farben kodiert. Die Unterüberschriften geben den Exponenten an, der in der jeweiligen Unterabbildung zum Skalieren der beiden Kurven genutzt wurde. Der Farbverlauf über die Unterabbildungen charakterisiert die Qualität des Datenkollapses (gelb  $\hat{=}$  niedrig, blau  $\hat{=}$  hoch) und wird so auch in Abbildung 3.5 und 3.6 genutzt. Der theoretische Exponent wurde in der letzten Unterabbildung verwendet.

Die einzelnen Kurven sind vergleichbar zu denen in Abbildung 3.3. Hier werden allerdings nur zwei Kurven mit  $N_1 = 2^{18}$  (durchgezogene Linie) und  $N_2 = 2^{19}$  (gepunktete Linie) zum Kollaps gebracht.

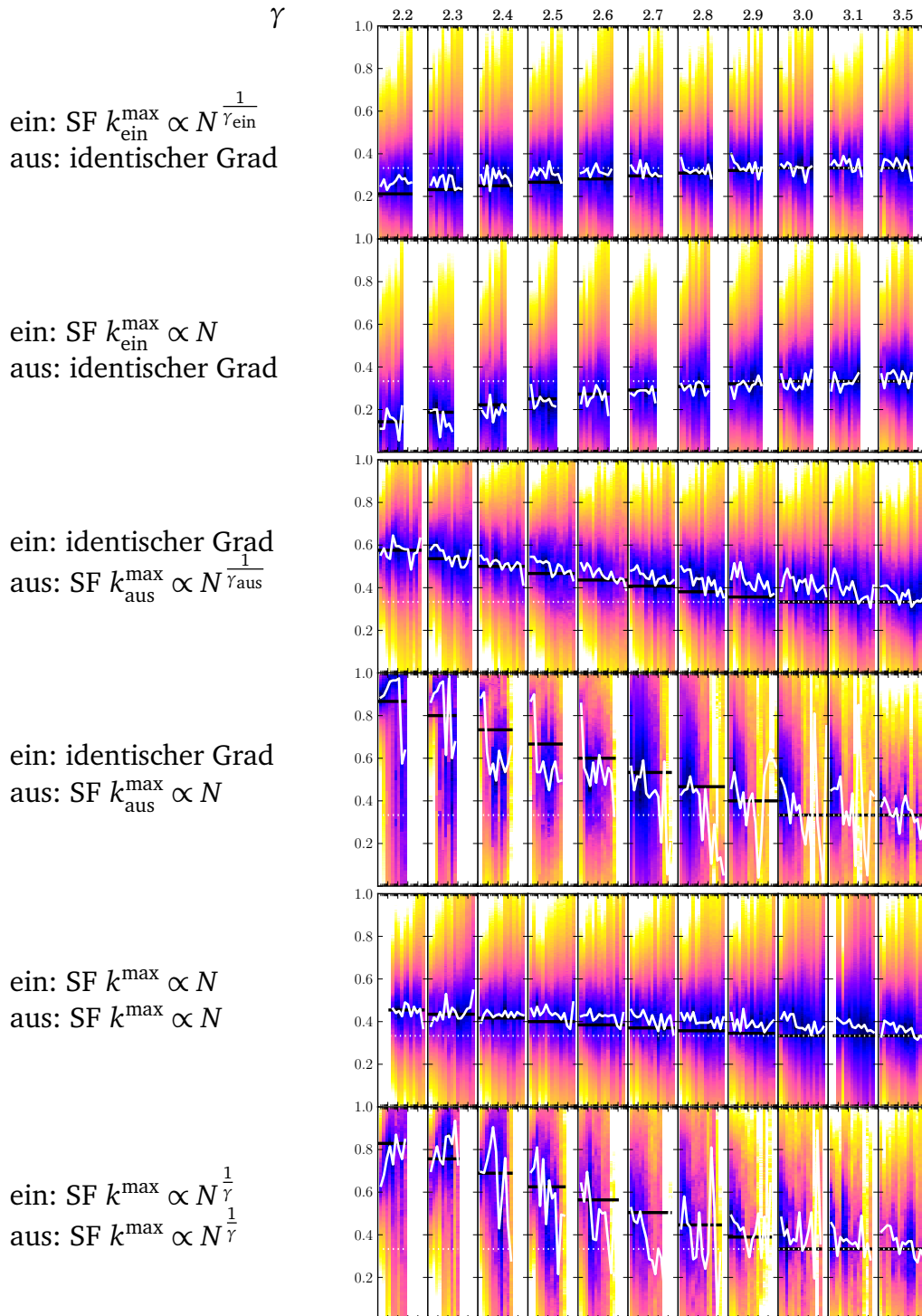




**Abbildung 3.5.: Numerische Ergebnisse zum Skalenexponenten  $x$**

Hier wird für  $x$  die Qualität möglicher Datenkollapse farbkodiert für die Verteilung  $P(N_{\text{nf}})$ , gewonnen aus Simulationen mit zwei verschiedenen Systemgrößen  $N_1$  und  $N_2 = 2 \cdot N_1$ , aufgezeigt. Der Wert von  $N_1$  wird auf der x-Achse (jeweils logarithmisch von  $10^3$  bis  $10^8$ ) und der für den jeweiligen Kollaps genutzte Exponent auf der y-Achse (linear von 0 bis 1). Die verschiedenen Plots zeigen verschiedene Kombinationen der Ein- und Ausgangsgradverteilungen (vertikal) und verschiedene Werte für  $\gamma$  (horizontal). Die Linien geben den Wert  $\frac{2}{3}$  (weiße gestrichelte Linie), den theoretisch vorausgesagten Exponenten (schwarze Linie) und den besten Exponenten für den Kollaps des Kurvenpaares (weiße durchgezogene Linie) an.





**Abbildung 3.6.: Numerische Ergebnisse zum Skalenexponenten  $\gamma$**

Hier wird für  $\gamma$  die Qualität möglicher Datenkollapse farbkodiert für die Verteilung  $P(|C_2|)$ , gewonnen aus Simulationen mit zwei verschiedenen Systemgrößen  $N_1$  und  $N_2 = 2 \cdot N_1$ , aufgezeigt. Der Wert von  $N_1$  wird auf der x-Achse (jeweils logarithmisch von  $10^3$  bis  $10^8$ ) und der für den jeweiligen Kollaps genutzte Exponent auf der y-Achse (linear von 0 bis 1). Die verschiedenen Plots zeigen verschiedene Kombinationen der Ein- und Ausgangsgradverteilungen (vertikal) und verschiedene Werte für  $\gamma$  (horizontal). Die Linien geben den Wert  $\frac{1}{3}$  (weiße gestrichelte Linie), den theoretisch vorausgesagten Exponenten (schwarze Linie) und den besten Exponenten für den Kollaps des Kurvenpaars (weiße durchgezogene Linie) an.

Darüber hinaus habe ich die Berechnungen auch für Ensembles mit biased Aktualisierungsfunktionen durchgeführt, sowie für Netzwerke, die anstelle von konstanten Gradverteilungen Poisson-Gradverteilungen genutzt haben. Auch hier konnte ich numerisch meine gefundenen Exponenten bestätigen. Leider benötigen diese Varianten um einige Größenordnungen größere Netzwerke, um vergleichbar wenig finite-size Effekte zu zeigen, was den numerischen Zugang etwas erschwerte, insbesondere wenn viele Parameterkombinationen über weite Bereiche von  $N$  überprüft werden sollten.

### 3.7 Implementierung

Wie in Abschnitt 3.6 bereits beschrieben, sollen, für verschiedene Kombinationen und Parameterisierungen der Ein- und Ausgangsgradverteilungen, Daten für eine Statistik zur Anzahl der nicht-gefrorenen Knoten eines Netzwerks gesammelt werden. Eine solche Anzahl liefert ein Durchlauf des in Abschnitt 3.3 beschriebenen Containeralgorithmus. Für eine gute Statistik benötige ich typischerweise mindestens  $10^4$  Durchläufe je Parameterkombination, besser aber  $10^5 - 10^6$ . Die Laufzeit eines einzelnen Durchlaufs skaliert super-linear mit  $N$ , was einige Optimierungen erfordert, um große Systeme zugänglich zu machen.

Bei der Laufzeitorientierung ist es wichtig, sich einen Überblick über die Laufzeitenanteile der einzelnen Programmabschnitte zu verschaffen, um die eigene Arbeitszeit auf die Optimierung der Programmabschnitte mit dem größten Zeitanteil fokussieren zu können. Dies ist mittels *kcachegrind*<sup>3</sup> geschehen. Andererseits sollte der Programmcode universell sein, um auf Änderungen in der Fragestellung reagieren zu können. Hierfür wurden einige Teilaspekte der Containermethode in Klassen abstrahiert. Objektorientierte Programmierung kann sich negativ auf die Performance auswirken. Daher habe ich darauf geachtet, dass (insbesondere virtuelle) Methodenaufrufe selten genug sind und sich nicht negativ auswirken. Einen Überblick über die wichtigsten Klassen verschafft Abbildung 3.7.

<sup>3</sup> <http://kcachegrind.sourceforge.net>

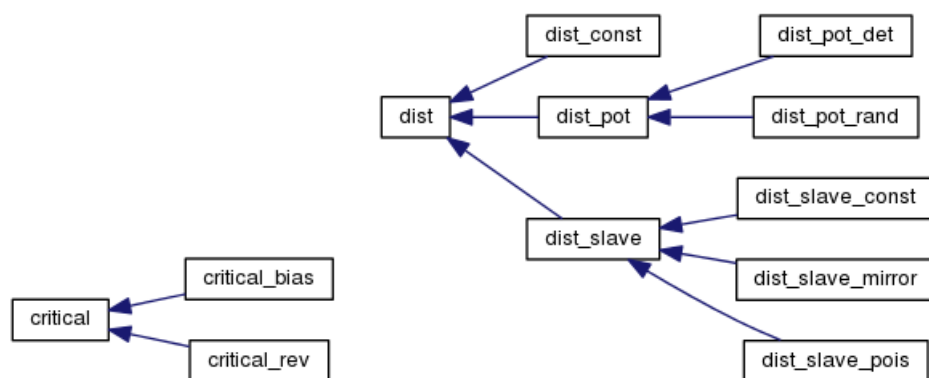


Abbildung 3.7.: Klassenhierarchie der wichtigsten Klassen, siehe Abschnitt 3.7

---

Eine dieser Klassen ist `dist`, zur Kapselung eine Verteilung. Diese liefert Stichproben von Verteilungen aus einem gegebenen Ensemble als C-Array. Durch Vererbung wurden hier unterschiedliche Typen für konstante Gradverteilungen (`dist_const`) und skalenfreie Verteilungen mit unterschiedlichem Cut-off Verhalten (`dist_pot_det` bzw. `dist_pot_rand`) erstellt. Diese können entweder als Ein- oder als Ausgangsgradverteilung genutzt werden. Die jeweils andere Gradverteilung muss exakt den gleichen Stichprobenumfang  $N$  sowie exakt den gleichen mittleren Grad  $K$  haben. Darum wird diese als *slave*-Verteilung gewählt: Eine *slave*-Verteilung bekommt eine andere Verteilung als Vorgabe (*master*) und imitiert in ihren Stichproben exakt den Mittelwert und den Stichprobenumfang der letzten Stichprobe der Masterverteilung. Hier habe ich drei Typen implementiert:

- `dist_slave_const` konstante Gradverteilung
- `dist_slave_mirror` Elemente der Stichprobe identisch, aber permutiert
- `dist_slave_pois` Poissonverteilung

Mein Programm lässt es offen, ob die *master*- oder die *slave*-Verteilung als Ein- bzw. Ausgangsgradverteilung dienen. In Simulationen nutze ich `dist_slave_mirror`, um sowohl Ein- als auch Ausgangsgradverteilung skalenfrei wählen zu können. In dem Fall sind beide Verteilungen vom gleichen Typ (Cut-off, sowie  $\gamma$ ), was eine starke Einschränkung bedeutet.

Nachdem eine Verteilung für die Eingangsgrade feststeht und die Knoten auf die Container verteilt wurden, müssen je nach Wahl der Booleschen Aktualisierungsfunktionen die Elemente der Container modifiziert werden, um ein kritisches Netzwerk zu erhalten (siehe Schritt 3. im Algorithmus aus Abschnitt 3.3). Dies geschieht, indem für konstante Aktualisierungsfunktionen die Knoten in Container  $C_0$  verschoben werden. Des Weiteren muss, ebenfalls abhängig von der Aktualisierungsfunktion, für einen Knoten mit  $n$  noch nicht gefrorenen Eingängen beim Einfrieren von  $i$  seiner Eingänge entschieden werden, ob er in Container  $C_0$  oder  $C_{n-i}$  verschoben wird (siehe Schritt 4. d)-g) im Algorithmus aus Abschnitt 3.3). Für die beiden untersuchten Typen der Aktualisierungsfunktion habe ich dies durch die Klassen `critical_bias` (biased Funktionen) und `critical_ref` (konstante und invertierbare Funktionen) implementiert.

Ein weiterer Trick zur Beschleunigung der Berechnung bestand darin, bei seltenen Booleschen Zufallsereignissen anstelle von einzelnen 1/0-Wahrscheinlichkeitsabfragen die Wahrscheinlichkeiten bis zur nächsten 1-Antwort zu berechnen und anhand dieser zu verfahren. Insgesamt war es mir möglich, Netzwerke bis zu einer Größe von  $\approx 2^{30}$  Knoten (limitiert durch den Arbeitsspeicher) zu untersuchen. Hier hat allerdings das Einfrieren (ein kompletter Durchlauf des Containeralgorithmus) eines einzelnen Netzwerks je nach Parametern Rechenzeit in der Größenordnung von Minuten (konstante Gradverteilungen) bzw. Tagen (skalenfreie Gradverteilungen) auf einem

---

Kern benötigt, so dass es hier nicht mehr möglich war, ausreichend Statistik (typischerweise  $10^4$ - $10^6$  Realisierungen) zu sammeln.

Da die von mir verwendeten skalenfreien Verteilungen sowohl sehr große als auch sehr kleine Wahrscheinlichkeiten enthalten, musste bei der Normierung von Wahrscheinlichkeiten vorsichtig verfahren werden. Hätte ich diese beginnend mit der größten Wahrscheinlichkeit aufsummiert, wären die kleinen Werte am Ende der Verteilung aufgrund der Beschränktheit der Mantisse (bei Double 52 Bit) ignoriert worden, obwohl sie aufgrund ihrer Anzahl einen bedeutenden Anteil haben. Um dies zu vermeiden, wurden Wahrscheinlichkeiten, wenn nötig, mittels des Kahan-Summationsalgorithmus [37] normiert.

Durch Precompiler-Abfragen wurde zwischen einer debugging-Version mit und einer performanten Produktivversion unterschieden. Wichtige Konsistenzprüfungen wurden aber auch in der Produktivversion mittels `assert(...)` aufgenommen, um auch selten auftretende Fehler finden zu können.

Um einzelne Experimente flexibel zu gestalten, wurden die in C++ geschriebenen Funktionen mittels Boost-Python in der Skriptsprache Python verfügbar gemacht. Aus Python heraus wurden einzelne Experimente gestartet und überwacht. Der gesamte Programmcode wurde mittels Versionsverwaltung (subversion) archiviert und versioniert. Jedes Experiment schreibt seine Ergebnisse in einen Ordner, dessen Pfad sich aus den Programmargumenten ableitet. Hier wird auch ausgegeben, welche Programmversion (subversion Revision) mit welchen Argumenten für die Ergebnisse in diesem Ordner verantwortlich war. So konnte auch bei sehr umfangreichen Experimenten gut überblickt werden, welche Ergebnisse aktuell und valide sind. Da ich alleine für einen Berechnungsversuch für die Daten aus Abbildung 3.5 über 1000 Parameterkombinationen testen musste, war es unabdingbar, die Rechnungen auf einem verteilten Cluster durchzuführen. Das Starten der einzelnen Berechnungen (Jobs) konnte ebenfalls nicht mehr manuell geschehen und wurde automatisiert. Zu einem Start gehörte auch das Sicherstellen, dass die genutzten Binärdateien aktuell sind, dem aktuellen Quellcode entsprechen und der Quellcode synchron mit den subversion repository ist. Zudem durften sich die Quellen bzw. Kompilate für eine Berechnungsserie während einer zum Teil mehrere Wochen dauernden Berechnung nicht zufällig ändern, obwohl ich diese aktiv weiterentwickelt habe. Auch das Zusammensetzen der Ergebnisse war aufwendig, da hier die Ergebnisse von (ggf. noch laufenden) Berechnungen mit teilweise identischen Parametern aber anderen Pseudozufallsgenerator-seeds gefunden und verarbeitet werden mussten.

---

### 3.8 Zusammenfassung

---

Mithilfe von analytischen Rechnungen, die durch Simulationen gestützt wurden, war ich in der Lage, sehr viele verschiedene Skalengesetze für die Anzahl der nicht-gefrorenen Knoten in kritischen skalenfreien Booleschen Netzwerken zu bestimmen. Meinen Berechnungen liegt ein Algorithmus zugrunde, der ausgehend von konstanten Funktionen den gefrorenen Kern iterativ bestimmt. Im Falle, dass die Exponenten der Potenzgesetze der Gradverteilungen kleiner 3 sind, hängen die gefundenen Exponenten hängen nicht nur von diesen ab, sondern auch davon, wie genau Stichproben aus dieser Verteilung erzeugt wurden. Die Skalengesetze wurden durch die Verallgemeinerung einer phenomenologischen Theorie gewonnen, die bereits früher für kritische Boolesche Netzwerke mit Poisson-Ausgangsgradverteilungen genutzt wurde. Darüber hinaus habe ich Simulationen mit Hilfe eines sehr effizienten Algorithmus durchgeführt, der es mir erlaubt, skalenfreie Netzwerke mit bis zu  $2^{28}$  Knoten zu untersuchen. Die durchgeführten Simulationen haben meine analytischen Überlegungen bestätigt. Diese Ergebnisse füllen eine wichtige Lücke im bisherigen Verständnis von kritischen Booleschen Zufallsnetzen.

Meine Resultate zeigen, dass die Größe des nicht-gefrorenen Teils des Netzwerkes mit wachsendem  $\gamma_{\text{ein}}$  anwächst, aber mit wachsendem  $\gamma_{\text{aus}}$  abfällt. Im Falle, dass der Cut-off der Ausgangsgradverteilung wie  $N$  skaliert, erreicht der Skalenexponent der nicht-gefrorenen Knoten 1, wenn sich  $\gamma_{\text{aus}}$  2 von oben annähert. Dies bedeutet, dass ein endlicher Teil des kritischen Netzwerkes im Grenzfall nicht-gefroren wäre. Im Fall  $\gamma_{\text{ein}} = \gamma_{\text{aus}} = \gamma$  hängt der Trend des Skalenexponenten vom Skalieren des Cut-offs mit der Netzwerkgröße ab. Entgegengesetzte Trends lassen sich für skalenfreie Ein- und Ausgangsgradverteilungen wie folgt erklären. Ein kleinerer Wert von  $\gamma_{\text{ein}}$  führt zu kleineren Fluktuationen im Inhalt der Container (d.h. im Wachstum des gefrorenen Kerns) und dadurch zu einem späteren Terminieren des Einfrieralgorithmus. Im Gegensatz dazu führt ein kleinerer Wert von  $\gamma_{\text{aus}}$  zu größeren Fluktuationen der Inhalte der Container und dadurch zu einem früheren Anhalten des Einfrieralgorithmus.

Der Anteil der nicht-gefrorenen Knoten mit zwei Eingängen wächst, wenn  $\gamma_{\text{aus}}$  in Richtung 2 verringert wird, und erreicht sogar den Wert 1 im Falle, dass der Cut-off der Ausgangsgradverteilung proportional zu  $N$  ist. Im Gegensatz dazu skaliert die Anzahl der nicht-gefrorenen Knoten mit zwei Eingängen im Falle von Poisson-Ausgangsgradverteilungen wie die Wurzel aus der Zahl der nicht-gefrorenen Knoten, unabhängig von der Eingangsgradverteilung. Für diesen Fall wurde in [38] gezeigt, dass der computational-core des Netzwerkes, welcher die Anzahl und Länge der Attraktoren bestimmt, ebenfalls mit diesem Faktor skaliert. Meine Ergebnisse zeigen, dass der computational-core wächst, wenn die Ausgangsgradverteilung skalenfrei mit einem Exponenten kleiner 3 ist und dass die Mehrheit der relevanten Elemente nicht mehr einfache Schleifen sind. Dies bedeutet, dass die Attrak-

---

toren im Falle von skalenfreier Ausgangsgradverteilung sehr viel länger sind als für skalenfreie Eingangsgradverteilungen.

Als letztes möchte ich darauf hinweisen, dass ich hier nur den Fall von unkorrelierten Ein- und Ausgangsgraden betrachtet habe. Doch auch der Fall von korrelierten Ein- und Ausgangsgraden (z. B. wenn diese identisch sind) ist relevant. Insbesondere gilt dies für Netzwerke mit ungerichteten Verbindungen, also solche, bei denen Kanten in beide Richtungen immer als Paar auftreten. Wie in [34] dargelegt, sollte der ungerichtete Fall mit einem Exponenten  $\gamma$  dem Fall von gerichteten Verbindungen mit Exponent  $\gamma - 1$  entsprechen.

---

### 3.9 Ausblick

---

Leider hätte die Untersuchung von doppelt skalenfreien Netzwerken mit Korrelationen zwischen Ein- und Ausgangsgrad den Rahmen dieser Arbeit gesprengt. Es gibt hierfür durchaus relevante Anwendungen. Der typische Fall ist sicher der, bei dem alle (oder viele) Knoten einen identischen Ein- und Ausgangsgrad haben. So etwas ist z. B. bei Booleschen Netzwerken mit ungerichteten Kanten der Fall. Ein Solches lässt sich auf ein Netzwerk mit gerichteten Kanten abbilden, wobei dafür Kanten immer paarweise (Hin- und Rückrichtung) auftreten müssten. Hier kommt also neben der Ein- und Ausgangsgradkorrelation zudem die Korrelation der Kanten erschwerend hinzu.

Zur Erweiterung der Containermethode auf solche korrelierten Ein- und Ausgangsgrade muss berücksichtigt werden, dass zwei Knoten in einem Container  $C_i$  sich nun womöglich dadurch unterscheiden, dass sie unterschiedliche Ausgangsgrade besitzen, die damit korrelieren, in welchem Container ein Knoten gestartet ist. Dies ließe sich zum einen durch einen zweiten Index an den Containern lösen (Containermatrix anstelle Containervektor). Da aber jeder dieser Indizes im schlimmsten Fall zwischen  $0, \dots, N$  liegen kann, würde ein solches Array  $4N^2$ -Byte an Speicher belegen (bei `int` Einträgen). Also wären bei 4 GByte an RAM nur maximal  $N \lesssim 15$  möglich, was meinen Erfahrungen nach vermutlich noch sehr starke finite-size Effekte zeigen würde. Eine solche Matrix wäre allerdings sehr dünn besetzt, da sich ihre  $N^2$  Einträge nur zu  $N$  aufsummieren. Sehr viel sinnvoller erscheint es zwei Vektoren der Länge  $N$  zu speichern, die zu jedem Knoten den Ein- und Ausgangsgrad enthalten. Diese Knoten können nun durch Sortierung und lookup-Tabellen zu *virtuellen* Containern zusammengefasst werden, wobei jeder Knoten hier die Information über seinen Ausgangsgrad von Anfang an zugeordnet bekommt und über den Prozess hinweg behält.



---

## 4 Blocknetzwerke

Der Inhalt dieses Kapitels beruht auf meiner gemeinsamen Arbeit mit Tiago Peixoto, die bereits in [39] veröffentlicht wurde.

---

### 4.1 Einleitung

---

In diesem Kapitel werde ich mich mit der Struktur von kritischen Netzwerken beschäftigen. Bisher besaßen alle Knoten in den betrachteten Netzwerken die gleichen Wahrscheinlichkeitsverteilungen für ihre Eingangs- und Ausgangsgrade sowie für die Wahl ihrer Aktualisierungsfunktionen. Dies werde ich in diesem Kapitel nun verallgemeinern und makrostrukturierte Boolesche Netzwerke untersuchen.

Einen guten Überblick über die Stärken und Schwächen einer Booleschen Dynamik von Netzwerken ist in [40] zu finden, insbesondere im Vergleich von Boolescher und kontinuierlicher Dynamik. Es hat sich gezeigt, dass spezielle boolesche Makrostrukturen notwendig sind, um eine Trajektorie eines Attraktors beispielsweise nach Störungen von außen immer noch zuverlässig zu durchlaufen. In [40] werden Makrostrukturen speziell auf ein solches Ziel hin optimiert. Diskutierte Optimierungskriterien sind unter anderem vergleichbares Verhalten von Boolescher und kontinuierlicher Dynamik [41, 42], Robustheit gegenüber von außen eingeführtem Rauschen [43, 10] oder Reproduzierbarkeit einer vorgegebenen Trajektorie [44].

Makrostrukturen von Graphen<sup>1</sup> als solches, d.h. ohne Boolesche Dynamik, sind in einem anderen Zweig der Forschung ebenfalls von starkem Interesse. Um ganze Ensembles von Graphen zu beschreiben, wurde das Konzept der stochastischen Blockmodelle [45] eingeführt. Diese wurden beispielsweise genutzt, um Graphen maximaler Robustheit gegenüber zufälliger oder gezielter Kantenentfernung zu konstruieren [46].

Ich werde im Folgenden keine Optimierungsziele für Makrostrukturen vorgeben, sondern einzig und allein die Kritikalität des Netzwerkes fordern. Alle weiteren Parameter sollen maximal zufällig gewählt werden. Mit anderen Worten werde ich anstelle einer Herangehensweise über mögliche Dynamiken oder evolutionäre Prozesse, die zu einem kritischen System führen können, hier die minimalen Eigenschaften in den Vordergrund stellen, die im Allgemeinen zur Kritikalität nötig sind. Mein Ziel ist es somit *Nullmodelle* herzuleiten, die die nötigen topologischen und funktionalen Charakteristiken zur Kritikalität aufweisen, aber abgesehen davon maximal zufällig sind. Ich bin hier sowohl an der makroskopischen Netzwerkstruktur als auch an der zugehörigen Wahl der Booleschen Funktionen interessiert.

---

<sup>1</sup> Es werden je nach Anwendung sowohl gerichtete als auch ungerichtete Netzwerke mit oder ohne Zufallseigenschaften untersucht.

Um Makrostrukturen in Booleschen Zufallsnetzwerken zu parametrisieren, werde ich im Folgenden das Konzept stochastischen Blockmodelle [45] um Boolesche Eigenschaften erweitern [43]. Diese Ensembles werden unter der Nebenbedingung der Kritikalität in ihrer Entropie [47] maximiert. Hierfür werde ich Kritikalitätsbedingungen für stochastische Blockmodelle aufstellen und diskutieren. Für sehr spezielle Makrostrukturen kritischer boolescher Netzwerke wurde dies schon in [48] durchgeführt.

Durch das Variieren sowohl des mittleren Grades des Netzwerkes als auch der relativen Gewichtung der entropischen Kosten von Struktur- und Funktionenensemble konnte ich topologische Phasenübergänge von einem normalen Zufallsnetzwerk zu vielen verschieden gearteten strukturierten Netzwerken hin beobachten. Auffallend waren hier z. B. eine core-periphery-artige Struktur, bei welcher die Dynamik einer Gruppe von Knoten durch eine andere Gruppe bestimmt wird (vgl. computational-core in Abschnitt 1.2.1.2) sowie eine Struktur aus zwei Gruppen von Knoten, bei der Knoten einer Gruppe nur eine sehr geringe Sensitivität aufweisen. Auch konnte ich zeigen, dass es unter allgemeinen Randbedingungen ausreicht, maximal zwei makroskopische Gruppen zu betrachten. Dies deutet darauf hin, dass kompliziertere Makrostrukturen sowie die zugehörige Wahl der Aktualisierungsfunktionen nicht zwingend direkt aus der Optimierung zur Kritikalität herrühren und möglicherweise andere Ursachen haben. Diese könnten z. B. das Ergebnis von Wachstumsprozessen oder anderer Nichtgleichgewichtsthermodynamiken sein, oder auch aus weiteren (ggf. verdeckten) Optimierungskriterien herrühren.

In Abschnitt 4.2 dieses Kapitels werde ich mein Modell präsentieren. In Kapitel 4.3 beschreibe ich die optimierten Ensembles der kritischen Netzwerke. In Abschnitt 4.5 stelle ich die Ergebnisse der Optimierung sowie das Phasendiagramm vor. Abschließend werde ich die Ergebnisse in Abschnitt 4.7 noch einmal in einer allgemeinen Diskussion zusammenfassen.

---

## 4.2 Das Modell

---

Um Makrostrukturen der Netzwerktopologie von kritischen Netzwerken zu parametrisieren, nutze ich allgemeine stochastische Blockmodelle [45, 47], bei denen die Knoten in  $B$  Gruppen unterteilt sind. Auch hier bezeichne ich die Gesamtanzahl an Knoten mit  $N$  und die Gesamtanzahl an Kanten mit  $E$ . Die Anzahl der Knoten in Gruppe  $r$  ist durch  $N_r = N \cdot n_r$  gegeben<sup>2</sup>. Die Matrix  $E_{r,s} = E \cdot e_{r,s}$  gibt die Anzahl der Kanten ausgehend von Knoten in Gruppe  $s$  zu Knoten in Gruppe  $r$  an. Abgesehen von diesen Anzahlen ist die übrige Struktur rein zufällig. In Abbildung 4.1 ist ein Beispiel für das Konzept der Blocknetzwerke zu sehen.

---

<sup>2</sup> Beachte, dass  $N_i$  im Zusammenhang der Containermethode zuvor anders definiert war.

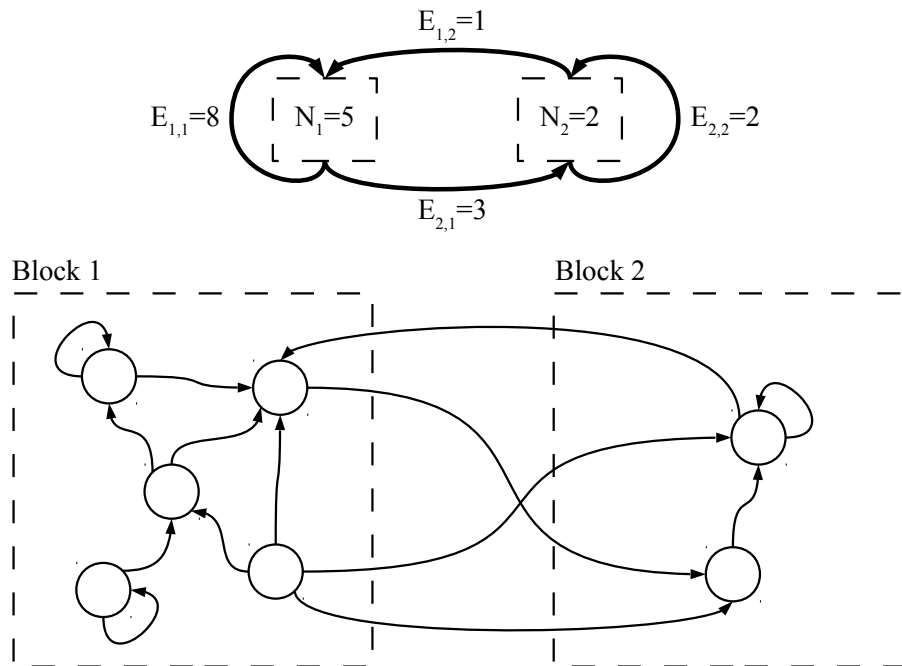


In diesem Kapitel werde ich von biased Aktualisierungsfunktionen ausgehen, siehe Abschnitt 1.1.2. Solche Funktionen besitzen einen Parameter  $p$ , den ich im Falle des Blockmodells für die Knoten jedes Blocks individuell mit  $p_r$  wählen werde.

#### 4.2.1 Kritikalität von Blocknetzwerken

Da ich mich hier für kritische Netze interessiere, muss ich die Definition der Hamming-Distanz (siehe Abschnitt 1.2.2) auf Blocknetzwerke erweitern. Sei die Hamming-Distanz zwischen zwei Zuständen eines Blocks  $r$  zum Zeitpunkt  $t$  mit  $h_r(t)$  gegeben, dann definiere ich die Hamming-Distanz  $h(t)$  für das gesamte Netzwerk als das gewichtete Mittel der einzelnen Distanzen

$$h(t) = \frac{1}{N} \sum_r n_r h_r(t). \quad (4.1)$$



**Abbildung 4.1.: Beispiel für ein gerichtetes Blocknetzwerk**

Hier ist exemplarisch ein stochastisches Blockmodell (obere Hälfte) sowie eine beispielhafte Realisierung (untere Hälfte) visualisiert. Das Ensemble ist parameterisiert

mit  $(E_{r,s}) = \begin{pmatrix} 8 & 1 \\ 3 & 2 \end{pmatrix}$  und  $(N_r) = \begin{pmatrix} 5 \\ 2 \end{pmatrix}$ .

Hieraus lassen sich folgende Größen ableiten:

$$N = \sum_r N_r = 7, E = \sum_{r,s} E_{r,s} = 14, K = \frac{E}{N} = 2, (e_{r,s}) = \left( \frac{E_{r,s}}{E} \right) = \begin{pmatrix} \frac{8}{14} & \frac{1}{14} \\ \frac{3}{14} & \frac{2}{14} \end{pmatrix},$$

$r$	$n_r = \frac{N_r}{N}$	$K_r^{\text{ein}} = \sum_s \frac{E_{r,s}}{N_r}$	$K_r^{\text{aus}} = \sum_{r'} \frac{E_{r',r}}{N_r}$	$K_r^{\text{selbst}} = \frac{E_{r,r}}{N_r}$	$K_r^{\text{kreuz}} = \frac{E_{3-r,r}}{N_r}$
1	5/7	9/5	11/5	8/5	3/5
2	2/7	5/2	3/2	1	1/2

Mithilfe von Gleichung (1.4) und Block-spezifischen Sensitivitäten  $\lambda_r = 2p_r(1 - p_r)$  lässt sich für den Vektor  $h_r(t)$  die Zeitentwicklung

$$h_r(t+1) \simeq \lambda_r \sum_s \frac{E_{r,s}}{N_r} h_s(t) \quad (4.2)$$

angeben, wobei  $\frac{E_{r,s}}{N_r}$  der Anteil der Eingänge von Gruppe  $r$  ist, die der Gruppe  $s$  entspringen. Die allgemeine Form dieses linearen Gleichungssystems ist

$$\vec{h}(t) = \Lambda^t \vec{h}(0), \quad (4.3)$$

wobei  $\vec{h}(t) = \{h_r(t)\}$  und  $\Lambda_{r,s} = \lambda_r \frac{E_{r,s}}{N_r}$  ist. Wenn  $\Lambda$  irreduzibel ist, lässt sich die Hammingdistanz mit Hilfe des Spektralradius<sup>3</sup>  $\varrho(\Lambda)$  nach oben durch

$$h(t) \leq \varrho(\Lambda)^t h(0) \quad (4.4)$$

abschätzen. Wenn  $\Lambda$  reduzibel ist, kann  $h(t)$  wegen  $\varrho(\Lambda) = 1$  höchstens polynomiell in  $t$  wachsen, was sich für große  $t$  stark von dem exponentiellen Wachstum im Falle von  $\varrho(\Lambda) > 1$  unterscheidet. Außerdem kann ein reduzibles  $\Lambda$  aufgrund der Stetigkeit des Spektralradius mittels einer minimalen Störung durch eine positive, irreduzible Matrix genähert werden.

Ein kritisches Netzwerk zeichnet sich dadurch aus, dass die Größe einer (beliebigen) Störung weder mit der Zeit anwächst noch abklingt. Dies bedeutet für Blocknetzwerke, dass der Spektralradius die Rolle der Sensitivität bei klassischen Booleschen Netzwerken einnimmt und ein Blocknetzwerk für  $\varrho(\Lambda) = 1$  kritisch ist. Im Folgenden werde ich  $\lambda := \varrho(\Lambda)$  für die Sensitivität des gesamten Blocknetzwerkes verwenden und  $\lambda_r$  für die Sensitivität des Blocks  $r$ .

Die Matrix  $\Lambda = (m_{i,j})$  ist aufgrund ihrer Konstruktion nicht-negativ ( $m_{i,j} \geq 0$ ). Für solche Matrizen lässt sich zeigen, dass jeder geschlossene Pfad der Länge  $l$

$$m_{i_1,i_2} m_{i_2,i_3} \cdots m_{i_l,i_1} \leq \varrho(\Lambda)^l \quad (4.5)$$

durch die  $l$ -te Potenz des Spektralradius nach oben beschränkt ist, siehe Abschnitt A.1. Speziell für die Diagonalelemente erhalte ich für kritische Netzwerke ( $\varrho(\Lambda) = 1$ )

$$\forall r : \lambda_r \frac{E_{r,r}}{N_r} \leq 1, \quad (4.6)$$

unabhängig davon, ob  $\Lambda$  reduzibel ist oder nicht. Vergleich mit der Kritikalitätsbedingung für konventionelle kritische Netzwerke (siehe Kapitel 1.2.2) zeigt, dass jeder individuelle Block (unter Vernachlässigung der Inter-Block-Kanten) kritisch oder gefroren sein muss. Jede zusätzliche Kante zwischen Blöcken kann die Situation in Richtung Chaos verschieben, solange hierbei

<sup>3</sup> der Betrag des betragsmäßig größten Eigenwertes

ein geschlossener Pfad (4.5) verstärkt wird. Insbesondere ist auch noch anzumerken, dass im Falle von  $B = 1$  Blöcken die Ergebnisse für klassische Netzwerke wegen  $\Lambda = (2p_1(1 - p_1)K)$  reproduziert werden [5].

---

#### 4.2.2 Kritikalität bei Netzwerken mit zwei Blöcken

---

Für ein System mit  $B = 2$  Blöcken kann ich die vorherigen Überlegungen noch verfeinern. Aus Gleichung (4.5) lässt sich in diesem Fall

$$\lambda_1 \frac{E_{1,2}}{N_2} \lambda_2 \frac{E_{2,1}}{N_1} \leq 1 \quad (4.7)$$

ableiten, vergleiche auch (A.11).

Des Weiteren kann ich die Eigenwerte  $\alpha_1$  und  $\alpha_2$  von  $\Lambda$  explizit angeben:

$$\alpha_{1/2} = \frac{\text{Tr } \Lambda}{2} \pm \sqrt{\left(\frac{\text{Tr } \Lambda}{2}\right)^2 - \det \Lambda}. \quad (4.8)$$

Durch die Bedingung  $\text{Tr } \Lambda = \lambda_1 \frac{E_{1,1}}{N_1} + \lambda_2 \frac{E_{2,2}}{N_2} \leq 2$  lässt sich sicherstellen, dass höchstens ein Eigenwert den Wert 1 annimmt. Da diese Bedingung schwächer als die der Gleichung (4.6) ist, ist sie redundant. Des Weiteren kann ich durch die Bedingung

$$\det(\Lambda - 1) = 0 \quad (4.9)$$

sicherstellen, dass es mindestens einen Eigenwert 1 gibt. Im Falle von zwei Blocknetzwerken ergibt dies

$$\left(E_{1,1} - \frac{N_1}{\lambda_1}\right) \left(E_{2,2} - \frac{N_2}{\lambda_2}\right) = E_{2,1} E_{1,2}. \quad (4.10)$$

Ich werde im Numerik-Abschnitt 4.4 auf Gleichung (4.10) zurückkommen, um meine Netzwerke möglichst kritisch ( $|\lambda - 1| \rightarrow 0$ ) zu halten und gleichzeitig mittels (4.38) einen Freiheitsgrad eliminieren zu können.

---

### 4.3 Optimierte Ensembles

---

Ich interessiere mich hier für die Erzeugung von Booleschen Netzwerken, die sich am kritischen Punkt  $\lambda = 1$  befinden. Um genauer zu sein, möchte ich *Nullmodelle* für kritische Netzwerke, die zusätzlich zur Kritikalität, bezüglich ihrer Netzwerkstruktur und der Funktionswahl, maximal zufällig sind. Dies erreiche ich, indem ich die Entropie  $S$  des betrachteten kritischen Ensembles maximiere. Diese Entropie setzt sich aus zwei Teilen zusammen, zum

Einen aus der Entropie des stochastischen Blockmodell-Ensemble [47]  $S_S$  (Strukturentropie)

$$S_S(\{n_r\}, \{e_{r,s}\}) = \ln \Omega(\{n_r\}, \{e_{r,s}\}) \quad (4.11)$$

mit der Gesamtanzahl der möglichen Netzwerke  $\Omega(\{n_r\}, \{e_{r,s}\})$ , zum Anderen aus der Entropie  $S_F$  der Verteilung der Booleschen Aktualisierungsfunktionen  $S_F(\{p_r\}, \{n_r\}, \{e_{r,s}\})$ .

Die Maximierung wird unter der Nebenbedingungen  $\lambda(\{n_r\}, \{e_{r,s}\}, \{p_r\}) = 1$  durchgeführt. Des Weiteren führe ich einen Parameter  $\mu \in [0, 1]$  ein, der die relative Gewichtung der entropischen Kosten zwischen einer Modifizierung der Netzwerkstruktur und der Booleschen Aktualisierungsfunktionen angibt. Die zu maximierende Größe ist damit

$$L = (1-\mu)S_S(\{n_r\}, \{e_{r,s}\}) + \mu S_F(\{p_r\}, \{n_r\}, \{e_{r,s}\}) - c \left| \lambda(\{n_r\}, \{e_{r,s}\}, \{p_r\}) - 1 \right|, \quad (4.12)$$

wobei  $c$  groß genug gewählt werden muss, um  $\lambda \approx 1$  sicherzustellen. Der Abstand  $|\lambda - 1|$  muss so lange verkleinert werden, bis eine weitere Verkleinerung keinen Einfluss mehr auf die Ergebnisse hat.

Für die Strukturentropie  $S_S(\{n_r\}, \{e_{r,s}\})$  kann ich nach [47]

$$S_S(\{n_r\}, \{e_{r,s}\}) \simeq E - \sum_{r,s} E_{r,s} \ln \left( \frac{E_{r,s}}{N_r N_s} \right) \quad (4.13)$$

$$= NK - NK \sum_{r,s} e_{r,s} \ln \left( \frac{NK e_{r,s}}{N^2 n_r n_s} \right) \quad (4.14)$$

$$= NK - NK \sum_{r,s} e_{r,s} \ln \left( \frac{K e_{r,s}}{n_r n_s} \right) + NK \ln \left( \frac{1}{N} \right) \quad (4.15)$$

nutzen, wobei vom Grenzfall von schwach verknüpften Netzwerken  $e_{r,s} \ll n_r n_s$  ausgegangen wurde. Dies entspricht aber ohnehin genau dem thermodynamischen Limes  $N^2 \gg E$ , von dem ich auch schon in Abschnitt 1.2.2 ausgegangen bin und bei dem die Wahrscheinlichkeit für Doppelkanten gegen 0 geht.

Als Entropie  $S_F$  für die Booleschen Aktualisierungsfunktionen habe ich drei unterschiedliche Varianten untersucht:

1. Zählen der möglichen Aktualisierungsfunktionen: Sei eine Menge von mit  $p_r$  parametrisierten biased Funktionen gegeben. Eine Boolesche Funktion mit  $i$  Eingängen würde eine Entropie von  $2^i H_b(p_r)$  zugeordnet, wobei  $H_b(x) = -x \ln x - (1-x) \ln(1-x)$  die binäre Entropiefunktion ist. Ohne weitere Einschränkungen sind die Eingangsgra-

de  $K_r$  in jedem Block poissonverteilt, welche nur durch den Mittelwert  $K_r^{\text{ein}} = \frac{\sum_s E_{r,s}}{N_r}$  parametrisiert ist.

$$S_F^T = N \sum_r \sum_i \frac{(K_r^{\text{ein}})^i}{i!} e^{-K_r^{\text{ein}}} n_r 2^i H_b(p_r) \quad (4.16)$$

$$= N \sum_r e^{K_r^{\text{ein}}} n_r H_b(p_r) \quad (4.17)$$

2. Dynamik: Wir können die Entropie des nächsten Zustandes eines Knotens verwenden, bei gegebenem Wissen über seinen aktuellen Zustand und unter der Voraussetzung, dass ein Zustandswechsel am Eingang stattfindet (eine Art Transinformation [49]). Diese Perspektive bezieht sich eher auf eine konkrete Dynamik und führt zu

$$S_F^B = N \sum_r n_r H_b(\lambda_r) \quad (4.18)$$

$$= N \sum_r n_r H_b(2p_r(1-p_r)) . \quad (4.19)$$

3. Lokal: Aus einer sehr lokalen Perspektive heraus lässt sich die Entropie des Zustandes eines Knotens verwenden, der eine zufällige Eingangs-konfiguration hat, was genau  $H_b(p_r)$  entspricht. Die gesamte Funktionsentropiefunktion wäre damit

$$S_F^P = N \sum_r n_r H_b(p_r) . \quad (4.20)$$

$H_b(p_r)$  und  $H_b(\lambda_r)$  können in Abbildung 4.2 verglichen werden.

Um bei der Maximierung der Gesamtentropie  $S = (1 - \mu)S_S + \mu S_F$  keine triviale Lösung zu erhalten, muss  $S_F$  nach oben (zumindest im Fall kritischer Netze) beschränkt sein, da auch  $S_S$  nach oben beschränkt ist. Andernfalls würden alle maximalen Lösungen für  $\mu > 0$  vollständig durch die Divergenz von  $S_F$  bestimmt und die Strukturentropie hätte keinen Einfluss mehr auf die Lösung.

Die Wahl  $S_F = S_F^T$  ist nicht nach oben beschränkt und somit als Wahl für  $S_F$  nicht geeignet. Folgender Fall ist ein typisches Beispiel für eine solche Divergenz: Ich nehme ein Netzwerk mit zwei Blöcken an, die jeder für sich kritisch sind und die keine Verbindungen untereinander haben. Um die Funktionsentropien  $S_F^T$  zu maximieren, reicht es aus, einen verschwindend kleinen, von 0 verschiedenen, Anteil des Netzwerkes<sup>4</sup> mit einer nicht verschwindenden Anzahl an Kanten innerhalb dieses Blocks zu konstruieren. Sei  $N_1 \ll E_{1,1}$  und

<sup>4</sup> Dies entspricht einem verschwindend kleinen Block

dadurch  $K_1^{\text{selbst}} = \frac{E_{1,1}}{N_1} \gg 1$ . In diesem Fall lässt sich die Kritikalitätsbedingung, die an Block  $r$  gestellt wird, umschreiben zu

$$1 \geq k_r \lambda_r = k_r 2p_r (1 - p_r) \approx k_r 2p_r, \quad (4.21)$$

womit sich die Entropie dieses Blocks  $S_F^T|_1$  schreiben lässt als

$$S_F^T|_1 = e^{K_1^{\text{ein}}} N_1 H_b(p_1) \quad (4.22)$$

$$= e^{\frac{E_{1,1}}{N_1}} N_1 H_b(p_1) \quad (4.23)$$

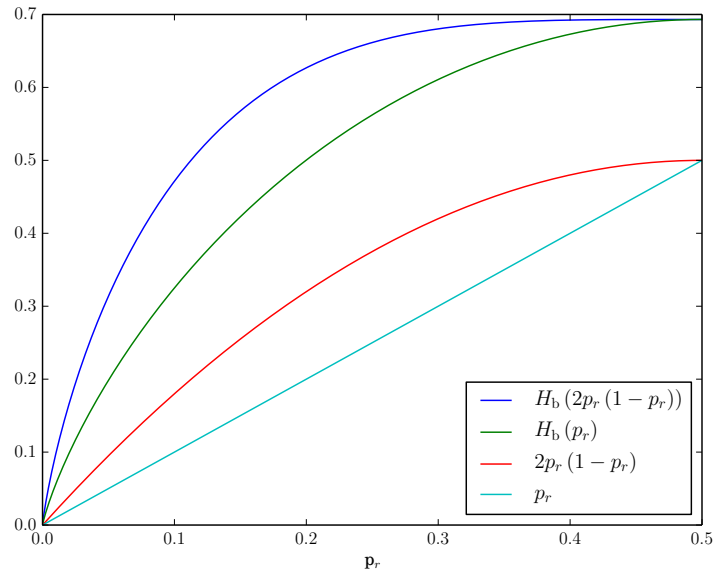
$$\approx e^{\frac{E_{1,1}}{N_1}} N_1 H_b\left(\frac{N_1}{2E_{1,1}}\right) \quad (4.24)$$

$$\approx e^{\frac{E_{1,1}}{N_1}} N_1 \frac{N_1}{2E_{1,1}} \ln\left(\frac{N_1}{2E_{1,1}}\right) \quad (4.25)$$

$$= e^{\frac{Ke_{1,1}}{n_1}} N n_1 \frac{n_1}{2Ke_{1,1}} \ln\left(\frac{n_1}{2Ke_{1,1}}\right). \quad (4.26)$$

Im Limes<sup>5</sup>  $n_1 \rightarrow 0$  bei konstanten  $e_{r,s}$  und  $K$  divergiert dieser Wert  $S_F^T|_1 \rightarrow \infty$ , obwohl das Gesamtnetzwerk immer noch kritisch ist und sich weder die Gesamtanzahl der Kanten noch der Knoten geändert hat. Die Ursache für dieses Problem ist die kombinatorische Explosion der möglichen Aktualisierungs-

<sup>5</sup> Der Block 1 muss mindestens einen Knoten enthalten. Da ich aber in dieser Arbeit im Allgemeinen nur am thermodynamischen Grenzfall  $N \rightarrow \infty$  interessiert bin, geht der Anteil dieses Blockes  $n_1$  bei jeder beliebigen konstanten Knotenzahl  $N_1 = n_1 N$  gegen 0.



**Abbildung 4.2.: Vergleich der unterschiedlichen Funktionsentropien  $S_F$**

Zusätzlich sind noch einige verwandte Größen gezeigt. Besonders die sehr schnell gegen 0 gehende Steigung von  $H_b(2p_r(1-p_r))$  für  $p_r \approx \frac{1}{2}$  hat Einfluss auf einen Teil der Ergebnisse.

funktionen mit steigendem Eingangsgrad. Diese wird nicht durch irgendeinen anderen Wert kompensiert, wie z. B. dem sinkenden Anteil von solchen Knoten. Zusätzlich käme hinzu, dass meine Grundannahme des schwach verknüpften Netzwerkes  $E \ll N^2$  innerhalb dieses Blocks nicht mehr gelten würde.

Darüber hinaus deuten empirische Studien an biologischen Systemen darauf hin, dass sich die meisten beobachteten Booleschen Aktualisierungsfunktionen in sehr schmalen Klassen aufhalten [50] und dass sich darüber hinaus die meisten möglichen Booleschen Funktionen entweder nicht biochemisch realisieren lassen oder nur sehr selten auftreten. Diese Wahl  $S_F = S_F^T$  der Funktionsentropie wurde auch aus diesem Grund nicht weiter von mir untersucht.

---

## 4.4 Numerische Methoden

---

Im Folgenden werde ich die Methoden beschreiben, mit denen ich die numerische Maximierung durchgeführt habe. Ziel ist es, die Funktion

$$(1 - \mu)S_S(\{n_r\}, \{e_{r,s}\}) + \mu S_F(\{p_r\}, \{n_r\}, \{e_{r,s}\}), \quad (4.27)$$

unter Variation von  $\{n_r\}$ ,  $\{e_{r,s}\}$  und  $\{p_r\}$  unter den Nebenbedingungen

$$\lambda(\{n_r\}, \{e_{r,s}\}, \{p_r\}) = 1 \quad (4.28)$$

$$\sum_r n_r = 1 \quad (4.29)$$

$$n_r \in [0, 1] \quad (4.30)$$

$$\sum_{r,s} e_{r,s} = 1 \quad (4.31)$$

$$e_{r,s} \in [0, 1] \quad (4.32)$$

$$p_r \in \left[0, \frac{1}{2}\right] \quad (4.33)$$

zu maximieren.

---

### 4.4.1 Nebenbedingungen

---

Der von mir gewählte Ansatz ist, die unbeschränkte Funktion

$$L = (1 - \mu)S_S + \mu S_F - \nu(\lambda - 1)^2 \quad (4.34)$$

zu maximieren, wobei  $\nu$  die Höhe der Bestrafung bestimmt, die eine Abweichung von  $\lambda = 1$  mit sich bringt. Für  $\nu \rightarrow \infty$  wird die originale, exakte Nebenbedingung angenommen. In der Praxis wähle ich  $\nu$  so groß, dass die Ergebnisse nicht weiter davon abhängen. Im weiteren Verlauf dieses Kapitels werde ich beschreiben, wie die Bedingung  $\lambda = 1$  dazu verwendet wird, die

Freiheitsgrade des Systems zu reduzieren. Leider ist dies nur bedingt möglich ( $B \leq 2$ ), was somit die Bestrafung für eine Abweichung von Nebenbedingung somit nicht immer unnötig macht. Als Heuristik für  $\nu$  hat sich

$$\nu = 10^5 \times \max(\max(S_S) - \min(S_S), \max(S_F) - \min(S_F)) \quad (4.35)$$

für konstantes  $K$  und variables  $\mu$  als brauchbar erwiesen. Ein konstanter Wert von  $\nu$  hat dagegen in Experimenten dazu geführt, dass der Wert von  $|\lambda - 1|$ , wenn auch sehr klein, stark von  $\mu$  und  $K$  abhing. Trotzdem konnte ich für ein konstantes  $\nu$  nicht ausschließen, dass dies zu Artefakten geführt hat.

Für die Berechnungen wurde  $N = 1'000$  festgesetzt, da aufgrund meines Ansatzes für  $S$  die optimalen Lösungen der Maximierung nicht von  $N$  abhängen. Die relevanten Terme meiner Zielfunktion sind linear von  $N$  abhängig für festes  $K$  und  $\mu$ . Alle weiteren Terme mögen andere Abhängigkeiten haben, sind aber nicht von den Variablen der Optimierung abhängig und bilden somit während der Optimierung (bei festen  $N, K, \mu$ ) eine irrelevante additive Konstante.

Um die Dimensionalität meines Lösungsraumes zu reduzieren und zudem einige Nebenbedingungen eliminieren zu können, habe ich noch weitere Überlegungen angestellt, die sich für  $n_i$  und für  $e_{r,s}$  verwenden lassen. Jedes  $L$ -Tupel von Parametern  $x_1, \dots, x_L$  mit  $1 = \sum_{i=1}^L x_i$  lässt sich aus einem anderen  $(L-1)$ -Tupel  $y_1, \dots, y_{L-1}$  ableiten, welches ebenfalls Werte in  $[0, 1]$ , aber keine Einschränkungen bezüglich der Summe hat. Ein simpler Ansatz hierfür wäre  $x_i = y_i$  für  $i < L$  und  $x_L = 1 - \sum_i y_i$ , welcher allerdings zu negativen Werten für  $x_L$  führen kann. Daher verwende ich einen aufwendigeren Ansatz, der zuerst die  $(y_i)$  als  $y'_1 \leq y'_2 \leq \dots \leq y'_{L-1}$  sortiert. Zusätzlich definiere ich  $y'_0 = 0$  und  $y'_L = 1$ . Die abschließende Transformation zu  $x_i$  ist durch  $x_i = y'_i - y'_{i-1}$  gegeben. Dieser Ansatz der Dimensionsreduktion lässt sich sowohl für  $n_i$  als auch für  $e_{r,s}$  nutzen.

Eine weitere Nebenbedingung ist  $\lambda = 1$ , welche sich selbst für  $B = 2$  nur schwer umformulieren lässt, da dies  $a_2 = 1$  (siehe (4.8)) erfordert. Viel einfacher und kompakter lässt sich der Umstand nutzen, dass  $\lambda$  ein Eigenwert von  $\Lambda$  ist<sup>6</sup>, was für den benötigten Spezialfall  $B = 2$  und  $\lambda = 1$  zu (4.10) führt. Dies lässt sich, um die zur Kritikalität erforderliche Sensitivität des zweiten Blocks explizit anzugeben, schreiben als

$$\lambda_2 = \frac{N_2}{\left( E_{2,2} - \frac{E_{2,1}E_{1,2}}{E_{1,1} - \frac{N_1}{\lambda_1}} \right)} \quad (4.36)$$

$$= 2p_2(1 - p_2) \quad (4.37)$$

$$p_2 = \frac{1}{2} - \sqrt{\frac{1}{4} - \frac{\lambda_2}{2}}. \quad (4.38)$$

<sup>6</sup> Im Falle von positivem  $\Lambda$  folgt dies aus dem Satz von Perron-Frobenius. Für nicht-negatives  $\Lambda$  lässt sich die Aussage, selbst im Falle von Reduzibilität, wegen der stetigen Abhängigkeit der Eigenwerte von den Matrixelementen durch Grenzübergang ableiten.



Wie man sieht, müsste  $p_2$  nach dieser Konstruktion in manchen Konstellationen außerhalb von  $[0, \frac{1}{2}]$  liegen bzw. wäre ggf. komplex oder sogar undefiniert. In meiner Numerik stelle ich sicher, dass die Blöcke individuell maximal kritisch sind, wie es in (4.6) verlangt wird bzw. ich wähle ein  $p_2$ , das die Anforderungen möglichst gut erfüllt (also typischerweise am Rand des Wertebereichs liegend), und überlasse es der Bestrafung mittels  $\nu$  bei Optimierung von (4.34) eine gültige Lösung zu erzielen. Diese Situationen sind nicht sehr häufig, treten aber während der Optimierung immer wieder auf und kommen sogar bei manchen optimalen Lösungen vor. Für  $B \leq 2$  führt dieser Ansatz aufgrund von (A.11) immer zu exakt kritischen Netzwerken, wie in Abbildung A.1 zu sehen ist.

Alle verbleibenden Nebenbedingungen der Optimierung sind Gültigkeitsintervalle der Variablen. Mit solchen *box constraints* können typische Optimierungsalgorithmen sehr gut umgehen, so dass hier keine Notwendigkeit besteht, diese durch weitere Transformationen auf gesamt  $\mathbb{R}$  abzubilden.

---

#### 4.4.2 Maximierung

---

Es hat sich im Laufe der Arbeit rasch gezeigt, dass eine Optimierung, die jede Kombination von  $K$  und  $\mu$  individuell optimiert, sehr stark verrauschte bzw. sogar falsche Ergebnisse liefert. Daher habe ich nur noch bei unterschiedlichen  $K$ -Werten unabhängig optimiert, bei  $\mu$  aber die erwartete Nachbarschaft von Lösungen für minimal unterschiedliche  $\mu$ -Werte genutzt. Auch dies führt jedoch zu Problemen an Phasenübergängen. Wenn ich eine Lösung, die für  $\mu$  optimal ist, als Startwert für eine Optimierung bei  $\mu + \delta\mu$  ( $\delta\mu \approx 10^{-4} \dots 7 \cdot 10^{-4}$ ) benutze, werden Phasenübergänge (oder ganze Phasen) oft übersehen bzw. am falschen Ort gefunden. Das Finden des Phasenübergangs am falschen Ort ließ sich dadurch beheben, dass ich für jedes  $\mu$  einen *aufsteigenden* und darauf einen *absteigenden* Optimierungslauf durchführte. Hier werden sowohl die als letztes angenommene Lösung für  $\mu$  bzw. die nächst kleinere / größere Lösung als Startwerte für eine verbesserte Lösung für  $\mu$  genutzt und das bessere der beiden Ergebnisse übernommen. Hierdurch werden ungenau gefundene Phasenübergänge nach oben oder unten korrigiert und somit der exakte Punkt bestimmt. Für die Optimierung wurde jeweils eine Kombination aus Powell's Algorithmus [51], gefolgt von einem downhill-Simplex [52] genutzt. Beide Verfahren benötigen eine Schrittweitenkonstante  $\varepsilon$ . Eine zu kleine Wahl führt dazu, dass der Algorithmus sehr nahe am Startwert verharrt, falls sich dort in der Nähe ein lokales Maximum befindet. Ein zu großes  $\varepsilon$  führt dazu, dass der genaue Wert des (globalen) Maximums nicht angenommen, sondern der Algorithmus bereits in seiner Nähe terminiert. Um dies zu umgehen, habe ich mehrere aufsteigende und absteigende Optimierungsläufe hintereinander geschaltet und jeweils dazwischen den Wert von  $\varepsilon$  um eine Größenordnung im Bereich von  $10^{-3}, 10^{-4}, \dots, 10^{-14}$  verkleinert.

All diese Ansätzen mit unterschiedlichen Optimierern für nur eine Zielfunktion waren leider nicht in der Lage, alle Phasen des Systems zuverlässig

zu entdecken. Speziell der Ansatz mit der aufsteigenden und absteigenden Optimierung kann Phasengrenzen verschieben, aber nur schwer neue finden. Um dies zu korrigieren, habe ich den Pareto-Optimierer S-Metric selection EMOA (SMS-EMOA) [53] eingesetzt, um zwei unabhängige Zielfunktionen maximieren zu können:

$$L_1 = S_S - v(\lambda - 1)^2 \quad (4.39)$$

$$L_2 = S_F - v(\lambda - 1)^2 \quad (4.40)$$

Um dem Algorithmus die Suche nach optimalen Lösungen zu erleichtern, wurden neben vielen zufälligen auch zwei bis drei bekannte optimale Lösungen für  $\mu = 0$  und  $\mu = 1$  in der Startpopulation (Populationsgröße 2000) benutzt. Des Weiteren wurde der Wert von  $v$  nicht direkt auf den maximalen Wert gesetzt, sondern zunächst mit einer schwächeren Bestrafung begonnen. Um optimale Ergebnisse für  $L$  zu bekommen, wurden für jeden untersuchten Wert von  $\mu$  alle Lösungen der Pareto-Front durchprobiert und die für das jeweilige  $\mu$  optimale übernommen. Die hieraus resultierenden Ergebnisse enthielten zwar alle Phasen, allerdings waren die Lösungen sehr unstetig. Diese konnten allerdings sehr gut mit dem oben vorgestellten Abstiegsverfahren verfeinert werden. Insgesamt ist meine Optimierung für festes  $K$  in Abbildung 4.3 beschrieben.

Um die Optimierung effizienter zu gestalten, habe ich die Aktualisierungsfunktionen anstelle von  $p_i$  direkt mit  $\lambda_i$  parametrisiert. Da es eine bijektive Abbildung auf dem relevanten Definitionsgebiet  $[0, \frac{1}{2}]$  zwischen diesen Größen gibt (4.38), ist dies problemlos möglich.

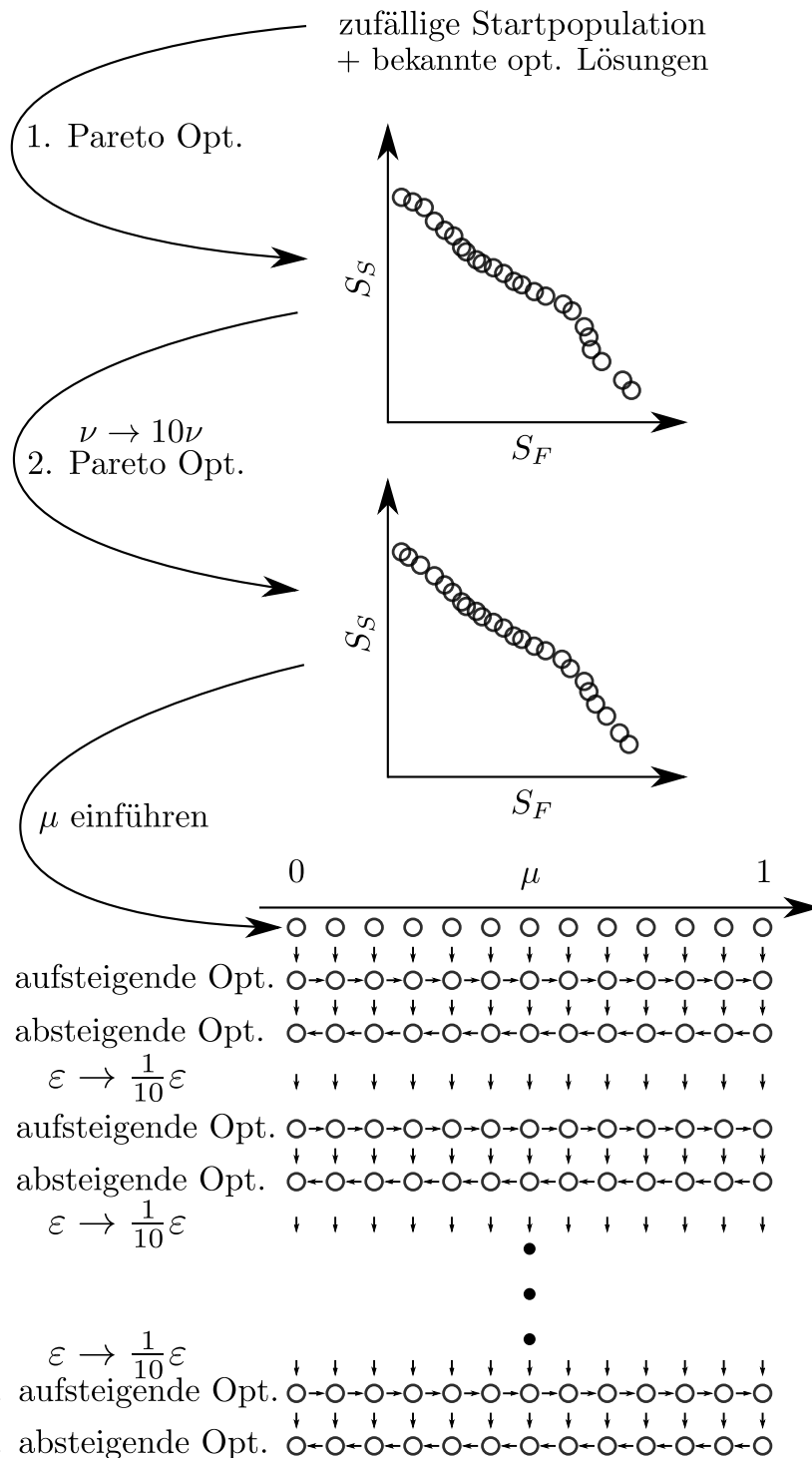
Der untersuchte Wertebereich für  $K$  war  $K - 2 \in [3 \times 10^{-6}, 10^3]$  in logarithmischen Schritten von  $\approx 1,26$ . Werte von  $K < 2$  ergeben keinen Sinn, da diese mit dem gewählten Aktualisierungsfunktionenensemble (biased Funktionen) nicht zu kritischen Netzwerken führen können. Für  $K = 2$  erfordert Kritikalität, dass alle  $p_r = \frac{1}{2}$  gewählt werden, und das Netzwerk maximal zufällig strukturiert ist. Somit ist dieser Fall auch nicht weiter von Interesse.

---

#### 4.4.3 Normierung

---

Nach meinem Modell sind die Blöcke untereinander ununterscheidbar. Dies führt in den Ergebnissen der Numerik dazu, dass sich benachbarte Lösungen (ähnliches  $K$  oder  $\mu$ ) nicht unbedingt ähnlich sehen, da ggf. Block 1 und 2 ihre Rollen tauschen. Dies lässt sich durch Einführen einer Anordnung der Blöcke nach einem *Sortierungskriterium* beseitigen. Für ein solches Kriterium gibt es sehr viele unterschiedliche Kandidaten. Simple Ansätze wie etwa  $n_i \leq n_{i+1}$  haben leider nicht dazu geführt, dass ausschließlich an wirklichen Phasenübergängen (siehe Kapitel 4.5) auch Unstetigkeiten in  $n_i$ ,  $e_{r,s}$ ,  $p_i$  oder daraus abgeleiteten Größen zu beobachten waren. Das beste von mir gefundene Sortierungskriterium ist eine Ordnung nach dem mittleren Eingangsgrad  $K_r^{\text{ein}} = \sum_s \frac{E_{r,s}}{N_r}$ , so dass in meinen Ergebnissen  $K_i^{\text{ein}} \leq K_{i+1}^{\text{ein}}$  gilt.



**Abbildung 4.3.: Schematischer Ablauf des Maximierungsalgorithmus für konstantes  $K$  und variables  $\mu$**

Jeder Kreis symbolisiert eine Lösung in einer Menge von Lösungen zu gegebenen  $K$ . In der unteren Bildhälfte deutet die horizontale Position eines Kreises den  $\mu$ -Wert der entsprechenden Lösung an. Knoten von denen kurze Pfeile ausgehen, werden als Startlösungen für Optimierungen verwendet. Falls zwei kurze Pfeile auf einen Kreis zeigen, bedeutet dies, dass beide Optimierungen durchgeführt werden, und das bessere Ergebnis verwendet wird.

Eine Lösung, die sich statistisch nicht von einer Lösung mit nur  $B = 1$  Blöcken unterscheiden lässt, ist typischerweise nicht direkt als solche erkennbar. Zur Identifizierung habe ich die Strukturentropie  $S_S$  als eine Art Signatur verwendet. Wann immer sich die Strukturentropie einer Lösung bis auf die Rechengenauigkeit nicht von der einer trivialen Lösung mit nur  $B = 1$  unterscheidet, habe ich diese Lösung durch die triviale  $B = 1$ -Lösung ersetzt.

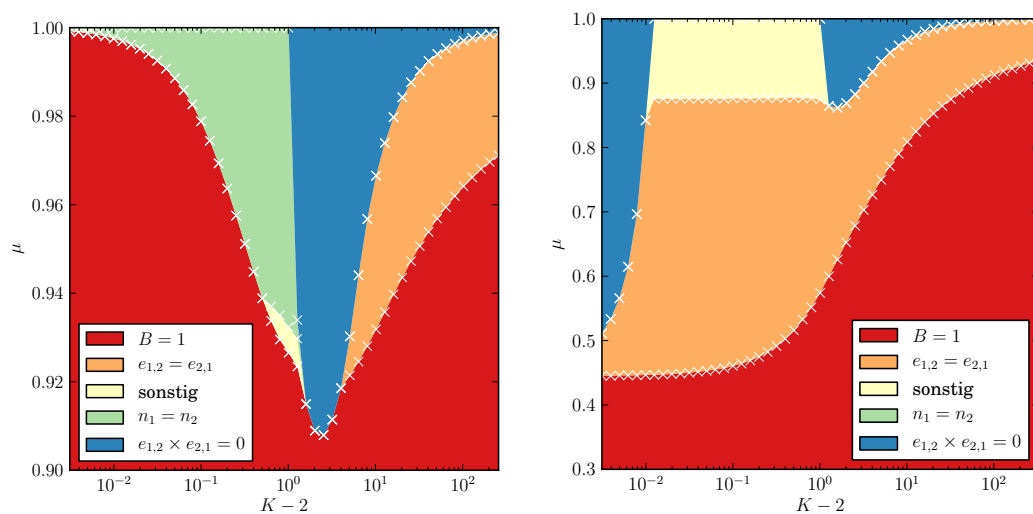
## 4.5 Numerische Ergebnisse

Ich habe im vorherigen Abschnitt beschrieben, wie ich Werte für  $\{n_r\}$ ,  $\{e_{r,s}\}$  und  $\{p_r\}$  erhalten habe, die die Funktion

$$(1 - \mu)S_S(\{n_r\}, \{e_{r,s}\}) + \mu S_F(\{p_r\}, \{n_r\}, \{e_{r,s}\})$$

unter der Nebenbedingung eines kritischen Netzwerkes  $\lambda = 1$  maximieren. Diese Ergebnisse sind jeweils für verschiedene Werte des mittleren Eingangs- bzw. Ausgangsgrades  $K = \frac{E}{N}$  und für unterschiedliche Gewichtungen  $\mu$  zwischen Struktur- und Funktionsentropie berechnet. Sie sind aufgrund vorheriger Annahmen in der Modellierung nur im thermodynamischen Limes  $N \gg 1$  gültig.

Ich allen von mir betrachteten Situationen reichte es aus, mit  $B = 2$  Blöcken zu rechnen. Denn jede Wahl von  $B > 2$  führte zu identischen Strukturen, bei den zwei oder mehrere Gruppen sich zusammenfassen ließen, ohne die Netzwerkstruktur zu verändern. Auch hier habe ich die Strukturentropie als Signatur verwendet (siehe auch Abschnitt 4.4.3).

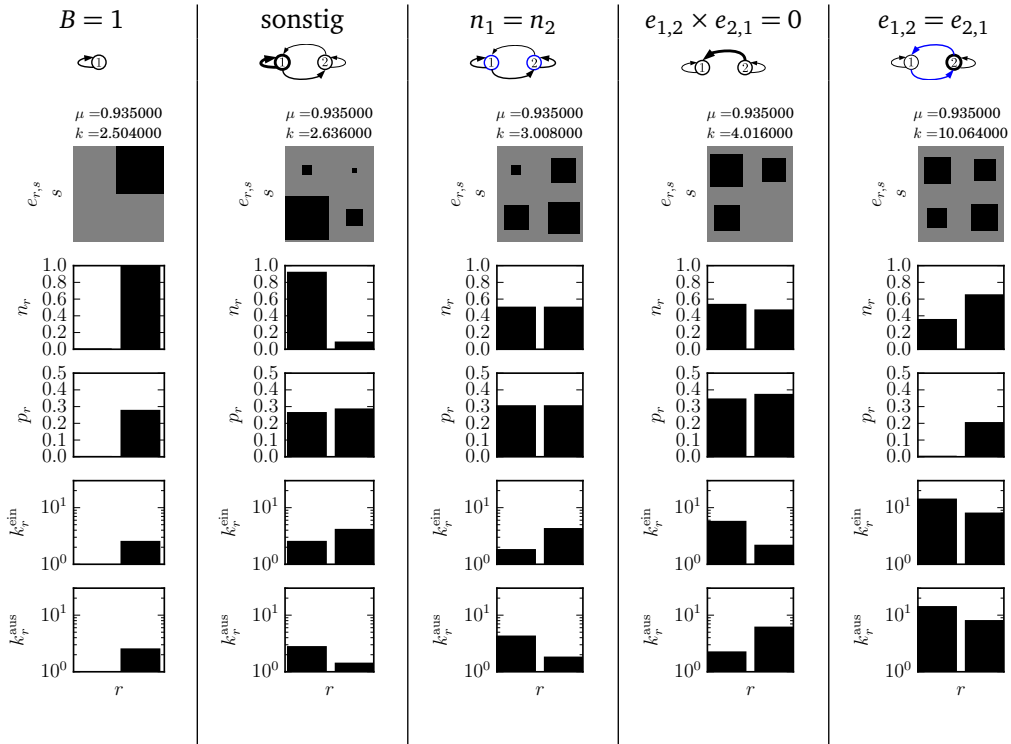


**Abbildung 4.4.: Phasendiagramm für kritisches Blocknetzwerk**

Links:  $S_F = S_F^B$ , Rechts:  $S_F = S_F^P$ . Die Phasen (siehe Legende bzw. Abbildung 4.5) wurden in Abhängigkeit des mittleren Eingangsgrades  $K$  und der Funktionsentropiegewichtung  $\mu$  angegeben.

Beim Variieren von  $K$  und  $\mu$  habe ich eine große Menge an Phasenübergängen gefunden. Die unterschiedlichen Strukturphasen ließen sich durch ihre deutlich ausgeprägten Makrostrukturen unterscheiden, wie in Abbildung 4.4 gezeigt. Beispiele für Zustände innerhalb der verschiedenen Phasen sind in Abbildung 4.5 gezeigt. Die Phasen werden weiter unten in diesem Abschnitt noch diskutiert. Ich habe die Phasen zunächst manuell durch Suche nach Unstetigkeiten in  $S_S, S_F, \{n_r\}, \{e_{r,s}\}, \{p_r\}, \dots$  bestimmt (Abbildung 4.6, A.2, A.3, A.4 und A.5), nach erfolgreicher Charakterisierung aber im Weiteren automatisch extrahiert. Hierfür war vor allem die in Abschnitt 4.4.3 beschriebene Normierung sehr wichtig, um soweit wie möglich nur echte Unstetigkeiten zu sehen, die sich nicht mittels Permutation der Blöcke eliminieren ließen.

Insgesamt kann ich für genügend kleine Werte von  $\mu$  beobachten, dass das Netzwerk eine komplett zufällige Phase annimmt, bei der  $B = 1$  Blöcke zur Beschreibung genügen würden. Der Parameter  $p$  der biased Aktualisierungsfunktionen wird in diesem Fall kritisch gewählt mit  $\lambda = 2p(1-p)K = 1$ . Diese Phase entspricht klassischen kritischen Booleschen Zufallsnetzwerken, die typischerweise in der Literatur [5] betrachtet werden. Sobald  $\mu$  genügend groß wird, und somit auch die entropischen Kosten, um eine sehr spezifische Bool-



**Abbildung 4.5.: Exemplarische, optimale Lösungen aus unterschiedlichen Phasen**  
aus Abbildung 4.4 für  $S_F = S_F^B$

Ganz oben ist eine grafische Repräsentation der Matrix  $e_{r,s}$  und des Vektors  $n_r$  gegeben, wobei die Dicke der Linien proportional zu  $e_{r,s}$  bzw.  $n_r$  ist. Blaue Linien weisen auf paarweise identische Werte hin. Darunter folgt eine Hinton-Darstellung der Matrix  $e_{r,s}$  und anschließend Histogramme der Vektoren  $n_r$ ,  $p_r$  sowie der mittleren Ein- und Ausgangsgrade  $K_r^{ein} = \sum_s \frac{E_{r,s}}{N_r}$  bzw.  $K_r^{aus} = \sum_{r'} \frac{E_{r',r}}{N_r}$  je Block.

sche Aktualisierungsfunktion zu wählen, stark steigen, erhalte ich Netzwerke mit einer gewissen Makrostruktur, die sich je nach genauem Wert von  $\mu$  und dem mittleren Grad  $K$  qualitativ unterscheiden. Für Werte von  $K \rightarrow 2^+$  erhalte ich für alle Werte von  $\mu$  ein komplett zufälliges Netzwerk ( $B = 1$ ) mit  $p = \frac{1}{2}$ , wie in [2]. Im Falle von  $S_F = S_F^P$  lässt sich dies nicht direkt erkennen, da die Phasengrenze zwischen  $B = 1$  und den anderen Phasen gegen einen konstanten, von 1 verschiedenen, Wert konvergiert. Genauere Betrachtungen zeigen jedoch, dass diese Phasengrenze einen kritischen Endpunkt bei  $K = 2$  hat und sich die Phasen für kleiner werdendes  $K$  quantitativ immer weiter annähern und im Grenzfall identisch sind.

Für Werte von  $K > 2$  wäre ein komplett zufälliges Netzwerk (Struktur und Funktionen) chaotisch. Daher entstehen andere Strukturen, die die Kritikalität sicherstellen.

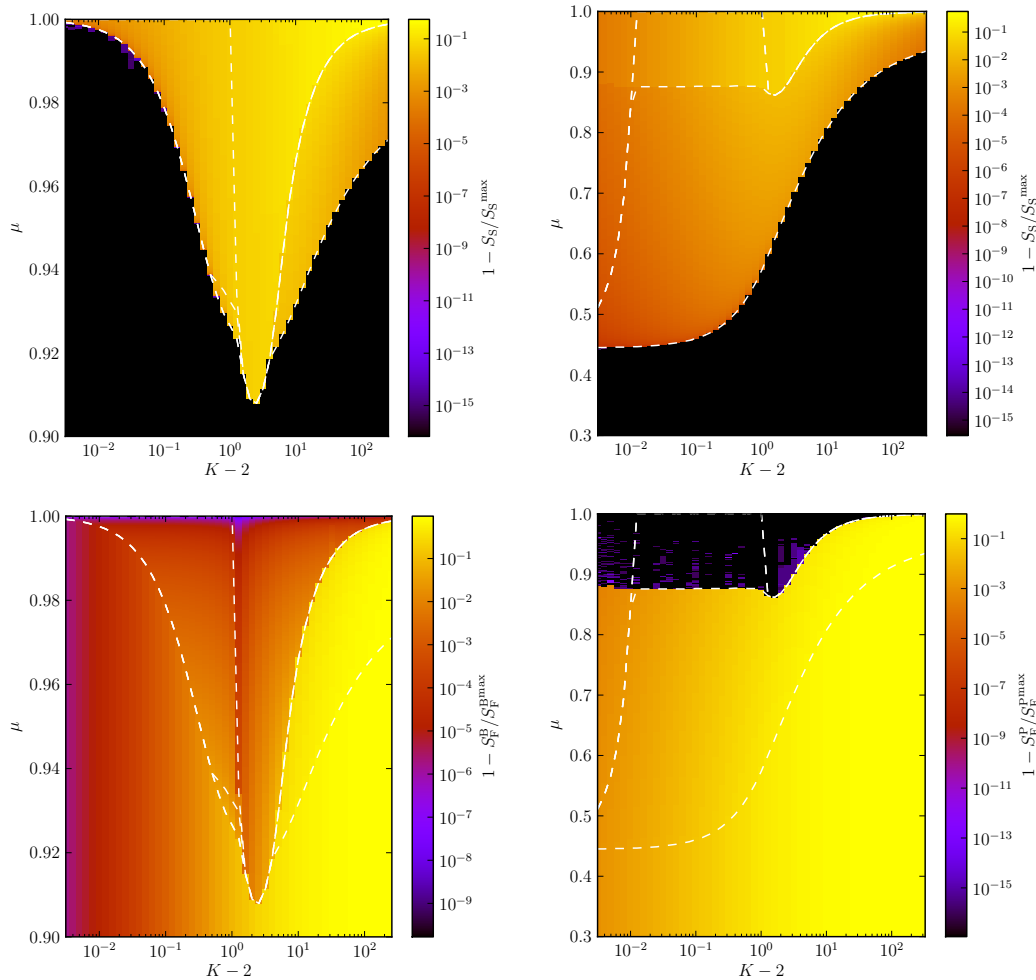
Im Fall von  $S_F = S_F^B$  existiert eine Region zwischen  $2 < K \lesssim 3$ , in der das System zwei Blöcke identischer Größe und Funktionen ( $n_1 = n_2$ ,  $p_1 = p_2$ ) bevorzugt, aber asymmetrische Verknüpfungen aufweist. Die *sonstig*-Phase wurde vermutlich aufgrund numerischer Ungenauigkeiten nicht ebenfalls automatisch als diese Phase klassifiziert. Rein optisch lässt sich hier in den Eigenschaften der optimalen Lösungen kein Phasenübergang erkennen. Für noch größere Werte von  $K$  entstehen zwei weitere neue Phasen. Die erste ist eine Art core-periphery Struktur ( $e_{1,2} \times e_{2,1} = 0$ ), bei der ein Block von einem anderen Block reguliert wird, aber nicht umgekehrt, zudem reguliert sich jeder Block selbst. Hierdurch können sich Störungen<sup>7</sup> nicht exponentiell zwischen den Blöcken aufschaukeln und bleiben somit klein, siehe auch Gleichung (4.5). Für noch größere Werte von  $K$  ist die Anzahl von Kanten zwischen beiden Blöcken identisch ( $e_{1,2} = e_{2,1}$ ), aber die Größe der Blöcke unterschiedlich und somit auch deren mittlerer Eingangsgrad. Der Block mit dem höheren Eingangsgrad hat einen sehr viel kleineren Wert von  $p_r$ , so dass seine Empfindlichkeit gegenüber Störungen signifikant kleiner ist als die des anderen Blocks.

Im Fall von  $S_F = S_F^P$  sieht das Phasendiagramm für Werte von  $K \gtrsim 10$  zu dem eben beschriebenen sehr ähnlich aus. Sie unterscheiden sich vor allem dadurch, dass es die drei dort existierenden Phasen hier auch für kleinere Werte von  $K$  gibt. Auffällig ist auch, dass die Phasengrenzen zwischen diesen für  $K \rightarrow 2^+$  gegen konstante von 1 verschiedene Werte konvergieren. Im Phasendiagramm sieht man die  $e_{1,2} \times e_{2,1} = 0$  Phase durch die *sonstig* bzw.  $n_1 = n_2$  Phase unterbrochen. Dies scheinen wie auch die *sonstig* Phase im vorhergehend diskutierten Fall numerische Ungenauigkeiten zu sein, die zu einer falschen automatischen Klassifizierung der Phase geführt haben. Auch hier lassen sich optisch keine Hinweise auf Phasenübergänge (z. B. Un-

<sup>7</sup> Die  $e_{1,2} \times e_{2,1} = 0$  Phase besitzt als einzige von mir gefundene Phase eine reduzierbare Sensitivitätsmatrix  $\Lambda$ . Entsprechend ist die in Abschnitt 4.2.1 besprochene Hammingdistanz im Falle eines kritischen Netzwerks nur polynomiell in  $t$  beschränkt. Eine minimale Störung von  $e_{r,s}$  würde den Charakter der Lösung nicht wesentlich verändern, aber zu einer konstanten oberen Schranke der Hammingdistanz führen.

stetigkeiten) in den Lösungen erkennen. Genauso verhält es sich mit der  $e_{1,2} \times e_{2,1} = 0$  Phase, die für kleine  $K$  die  $e_{1,2} = e_{2,1}$  verdrängt.

In Abbildung A.1 kann man sich davon überzeugen, dass die von mir gefundenen Lösungen bis auf sehr kleine numerische Ungenauigkeiten alle exakt einem kritischen Booleschen Netzwerk entsprechen.

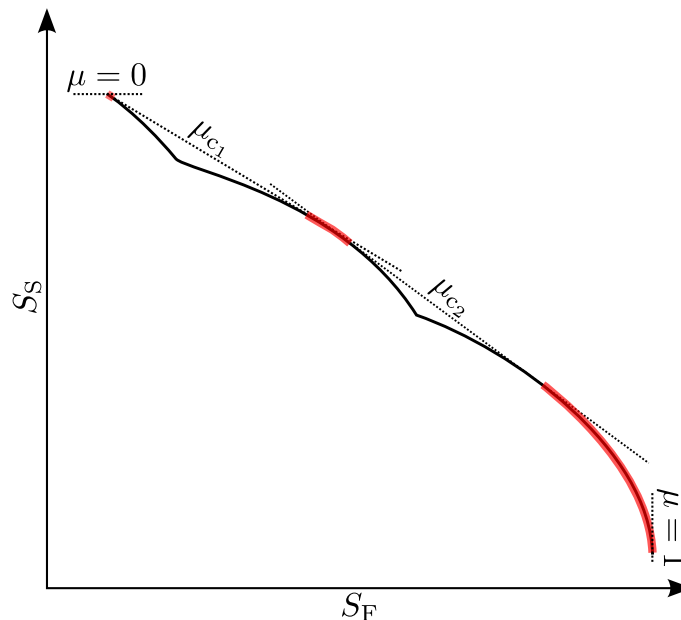


**Abbildung 4.6.: Relativer Abstand der Strukturentropie  $S_S$  (oben) und Funktionsentropie  $S_F^B$  (unten) von ihrem jeweiligen, maximalen Wert für jeweiliges  $K$**

Nur durch die relative Skalierung ist es möglich, manuell die Phasengrenzen zu erkennen, da ansonsten die relevanten (kleinen) Unstetigkeiten durch das Anwachsen von  $S$  mit  $K$  verdeckt werden würden. Links:  $S_F = S_F^B$ , Rechts:  $S_F = S_F^P$ . Die weißen Linien verdeutlichen die gefundenen Phasengrenzen.

## 4.6 Ursprung der Phasenübergänge

Um den Ursprung der Phasenübergänge zu verstehen, ist es nützlich die Pareto-Fronten der gemeinsamen Optimierung beider Entropiefunktionen  $S_S$  und  $S_F$  zu betrachten. Eine Pareto-Front ist eine Linie in der  $(S_S, S_F)$ -Ebene, für die es nicht möglich ist, einen der beiden Werte zu verbessern ohne den anderen zu verschlechtern. Da ich beide Werte linear zu  $L = (1 - \mu)S_S + \mu S_F$  kombiniere, ist auf jeder Linie in dieser Ebene mit einer Steigung  $\frac{\mu}{\mu-1}$  der Wert von  $L$  identisch. Der Maximierungsprozess besteht nun daraus, für einen gegebenen  $\mu$ -Wert den Schnitt der Front mit einer Gerade der Steigung  $\frac{\mu}{\mu-1}$  zu finden, die maximalen Abstand zum Ursprung hat (siehe Abbildung 4.7). Nicht alle Punkte auf der Pareto-Front können so zu einer optimalen Lösung für ein beliebiges  $\mu$  werden, sondern nur solche, die sich auf der konvexen Hülle der Front befinden. Eine Maximierung liefert oft nicht-zusammenhängende Bereiche der Pareto-Front. Der Übergang zwischen diesen Bereichen entspricht einem Phasenübergang erster Ordnung, da die Gesamtentropie  $S$  stetig ist (vergleichbar der freien Energie in  $F = E - TS$ ), ihre Komponenten  $S_S$  und  $S_F$  aber Unstetigkeiten aufweisen (vergleichbar Energie und Entropie in  $F = E - TS$ , die Temperatur  $T$  würde bei mir  $T \sim \frac{\mu}{\mu-1}$  entsprechen). In Abbildung 4.8 sind die konvexen Hüllen der Paretofronten



**Abbildung 4.7.: Qualitatives Beispiel einer Pareto-Front**

Diese Pareto-Front entsteht durch die gemeinsame Optimierung von  $S_S$  und  $S_F$  (schwarz). Die in rot hervorgehobenen Liniensegmente sind die konvexe Hülle der Pareto-Front, die einige beispielhafte Kurven mit Steigung  $\frac{\mu}{\mu-1}$  in größtmöglicher Entfernung zum Ursprung schneiden. In Bereichen, die nicht rot hervorgehoben sind, finden Phasenübergänge erster Ordnung statt, da hier die beiden Entropiekomponenten eine Unstetigkeit haben. Solche Bereiche lassen sich durch Steigungen charakterisieren, die sich jeweils in ein kritisches  $\mu_{c_i}$  umrechnen lassen.



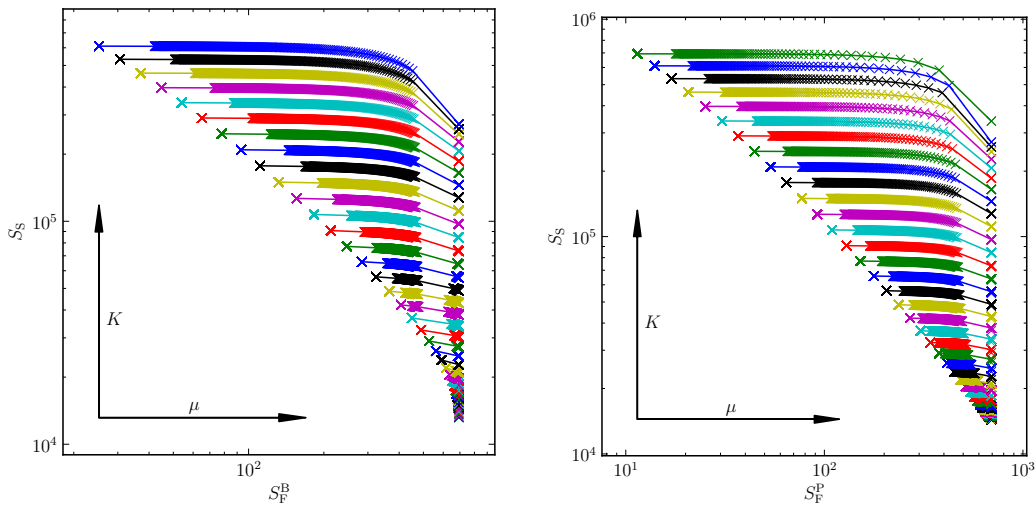
ten von all meinen Optimierungen abgebildet. Hier lassen sich sehr gut die Phasenübergänge als Lücken in den Fronten erkennen.

Des Weiteren konnte ich ein relativ einfaches quantitatives Modell für den Übergang von der komplett zufälligen Phase  $B = 1$  zu anderen, strukturierten Phasen finden. Für einen gegebenen Wert von  $K$  liegen beide Entropiewerte in gewissen von  $\mu$  abhängigen Intervallen, z. B.  $S_S \in [S_S^{\min}, S_S^{\max}]$  und  $S_F \in [S_F^{\min}, S_F^{\max}]$ . Dies lässt sich nutzen, um meine Zielfunktion umzuschreiben zu

$$L = (1 - \mu) (S_S - S_S^{\min}) + \mu (S_F - S_F^{\min}) + \text{konst.} \quad (4.41)$$

Für kleine Werte von  $\mu$  ist der Wert von  $L$  dominiert durch die Strukturentropie  $S_S$ , wohingegen bei großen Werten von  $\mu$  die Funktionsentropie  $S_F$  dominiert. Eine plausible Hypothese für die Position  $\mu_c$  des Phasenübergangs zwischen  $B = 1$  und einer anderen Phase ist, dass dies dann stattfindet, wenn beide Entropien mittels ihrer Gewichtung in etwa den gleichen Einfluss auf die Zielfunktion haben. Aus

$$(1 - \mu_c) \Delta S_S = \mu_c \Delta S_F \quad (4.42)$$



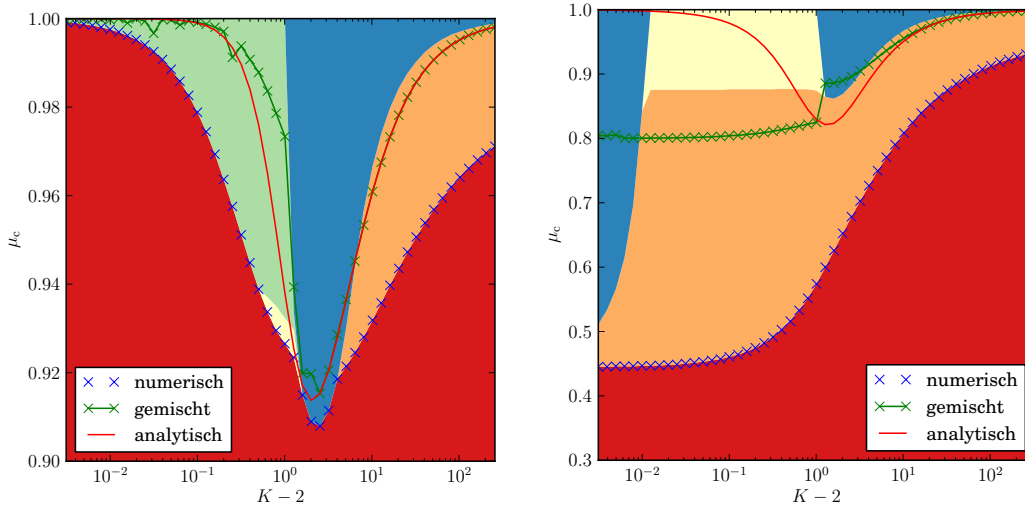
**Abbildung 4.8.: Konvexe Hüllen der Pareto-Fronten aller gefundenen Lösungen**  
Links:  $S_F = S_F^B$ , Rechts:  $S_F = S_F^P$ . Die angegebenen Entropien beziehen sich auf Ensemble mit  $N = 1'000$  Knoten, wobei andere Knotenzahlen lediglich die Werte auf den Achsen affin-linear transformieren würden. Die unterschiedlichen Farben stehen für unterschiedliche Werte von  $K$ , wobei Kurven weiter oben ein höheres  $K$  als Kurven weiter unten aufweisen. Der Wert von  $\mu$  nimmt innerhalb jeder Kurve von links nach rechts zu. In dieser Darstellung sind die Phasenübergänge sehr schön an den Bereichen ohne Datenpunkte zu erkennen. Es fällt auch auf, dass manche Phasen nur aus einem Datenpunkt zu bestehen scheinen. Diese Phasen zeichnen sich typischerweise dadurch aus, dass es in ihnen nur sehr ähnliche bzw. genau einen möglichen Zustand gibt, wie auch in Abbildung A.2 bzw. Abbildung A.4 zu sehen ist.

lässt sich der kritische Punkt

$$\mu_c = \frac{\Delta S_S}{\Delta S_S + \Delta S_F} \quad (4.43)$$

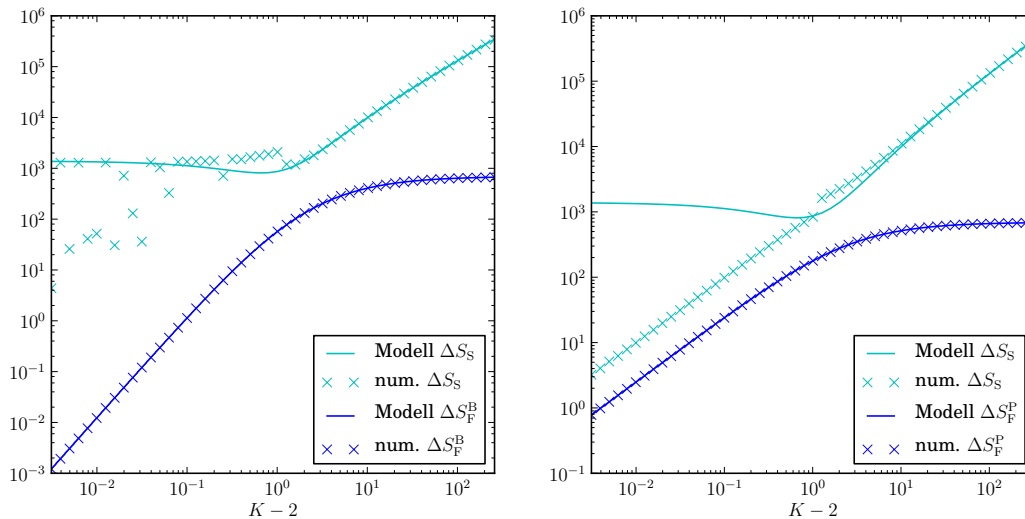
ableiten. Die Werte für  $\Delta S_S := S_S^{\max} - S_S^{\min}$  und  $\Delta S_F := S_F^{\max} - S_F^{\min}$  lassen sich entweder numerisch bestimmen oder durch einfache Argumente analytisch abschätzen. Die maximale Strukturentropie und minimale Funktionsentropie wird im komplett zufälligen Netzwerk mit  $B = 1$  und  $p_r = \frac{1}{2} - \sqrt{\frac{1}{4} - \frac{1}{2K}}$  angenommen. Die minimale Strukturentropie und maximale Funktionsentropie wird nach meiner Beobachtung in der  $e_{1,2} \times e_{2,1} = 0$  Phase angenommen. Deren Extremfall ist  $n_1 = n_2 = \frac{1}{2}$ ,  $e_{2,1} = 0$ ,  $e_{1,1} = e_{2,2} = \frac{1}{K}$ ,  $e_{1,2} = 1 - \frac{2}{K}$  und  $p_1 = p_2 = \frac{1}{2}$ . Wie in Abbildung 4.9 zu sehen ist, verhält sich der modellierte Wert von  $\mu_c$  sehr ähnlich zum numerischen. Einzig für  $K \lesssim 3$  gibt es im Falle  $S_F = S_F^P$  einen größeren Modellierungsfehler. In Abbildung 4.10 lässt sich der Grund hierfür gut erkennen. Dies liegt an einer falschen Annahme für die Lösung mit maximaler Funktionsentropie und minimaler Strukturentropie, wobei dies nur auf die maximale Funktionsentropie einen größeren Einfluss hat. Dieser Modellierungsfehler besteht zwar für beide Arten der Funktionsentropie, wirkt sich aber lediglich bei  $S_F = S_F^P$  stark aus, da hier das echte  $\Delta S_F^B$  über mehrere Größenordnungen ziemlich genau ein Vielfaches von  $\Delta S_S$  ist, was auch die gerade verlaufende Phasengrenze für  $K \rightarrow 2^+$  im linken Teil von Abbildung 4.4 erklärt. Bei  $S_F = S_F^B$  divergiert dieses Verhältnis im Modell und der Numerik unterschiedlich schnell, was aber zu einer fast identischen Phasengrenze führt. Leider war es mir nicht möglich, einen analytischen Ausdruck für den echten extremalen Zustand zu finden. Es scheint sich hier um eine Art  $B = 1$  Netzwerk mit einer minimalen Störung der Struktur zu handeln.

An dem in Abbildung 4.9 gezeigten Modell fällt darüber hinaus auf, dass der Wert des modellierten Phasenübergangs am Besten zu einem Phasenübergang zwischen zwei strukturierten Phasen passt und, nicht ganz so gut wie erhofft, zum Übergang zwischen der  $B = 1$  Phase und einer strukturierten Phase. Dies zeigt, was auch schon im unteren Teil von Abbildung 4.6 und Abbildung 4.8 zu erkennen war: Die  $e_{1,2} = e_{2,1}$  Phase zeigt eine immer noch sehr hohe Strukturentropie und ist somit eher einer Strukturentropie dominierten Phase als einer Funktionsentropie dominierten zuzuordnen.



**Abbildung 4.9.: Modelle für den kritischen Punkt  $\mu_c$ , des Übergangs zwischen komplett zufälliger Phase  $B = 1$  und einer strukturierten Phase**

Links:  $S_F = S_F^B$ , Rechts:  $S_F = S_F^P$ . Die *analytisch*-Kurve gibt das rein analytische Modell (4.43) für diesen Übergang an, die *gemischt*-Kurve das Modell mit numerischen Werten für  $\Delta S_S$  und  $\Delta S_F$  und die *numerisch*-Kurve den numerisch gefundenen Übergang. Es fällt auf, dass die beiden Modellkurven einige Charakteristika sehr gut wiedergeben. Überraschenderweise fallen sie eher mit einem anderen Phasenübergang zusammen als dem eigentlich modellierten.



**Abbildung 4.10.: Maximaler Unterschied in der Struktur- oder Funktionsentropie für festes  $K$  und beliebiges  $\mu \in [0, 1]$  in Abhängigkeit von  $K$**

Links:  $S_F = S_F^B$ , Rechts:  $S_F = S_F^P$ . Die *Modell*-Kurven entstehen aus Annahmen für die beiden Fälle, an denen die minimalen und maximalen Entropien angenommen werden. Die *num.*-Kurven entsprechen den Ergebnissen der numerischen Optimierung. Die *Modell* und numerischen Kurven liegen im Falle von  $\Delta S_F$  exakt aufeinander. Für  $\Delta S_S$  fällt auf, dass bei  $K \approx 3$  beide numerischen Kurven stark vom Modell abweichen und im Falle von  $S_F = S_F^B$  verrauscht sind.

---

## 4.7 Zusammenfassung

---

Ich habe in diesem Kapitel das Konzept der kritischen Booleschen Blocknetzwerke diskutiert und konnte sowohl eine Kritikalitätsbedingung ableiten als auch hieraus Kriterien für kritische Blocknetzwerke gewinnen. Zudem habe ich unterschiedliche Möglichkeiten aufgezeigt, Entropie für solche Ensemble zu definieren, und diese anschließend für den Fall der maximalen Entropie untersucht. Solche maximalentropischen Blocknetzwerke zeigen sehr viele Phasenübergänge unter Variation ihrer Dichte und der Gewichtung der Struktur- zur Funktionsentropie. Diese erstrecken sich von einer komplett zufälligen Phase zu mehreren unterschiedlichen mit Makrostrukturen.

Im Grenzfall mit hohen Kosten für nicht zufällige Boolesche Aktualisierungsfunktionen entsteht eine Zwei-Block-Topologie. Für sehr hohe Kosten ist dies eine Art core-periphery Struktur, wo ein Teil des Netzwerks mehr für die Regulation verantwortlich ist als der andere Teil. Für nicht ganz so hohe Kosten entstehen zwei gleich große Gruppen mit identischer Anzahl an Verknüpfungen in beide Richtungen zwischen den Blöcken, aber sehr unterschiedlicher Sensitivität. Falls die entropischen Kosten für die Netzwerkstruktur dominieren, entsteht ein komplett zufälliges Netzwerk mit identischem Funktionsensemble, das den klassischen Kritikalitätsbedingungen genügt.

Die entstandene core-periphery-artige Topologie ist ähnlich zu derjenigen, die bei der Optimierung gegen stochastische Fluktuationen [43] entsteht. Auch bei einer Optimierung auf strukturelle Robustheit gegen zufällige Ausfälle [46] findet sich diese wieder. Diese core-periphery-artige Struktur ist qualitativ ähnlich zu dem, was in echten genregulatorischen Netzwerken beobachtet wird, bei denen eine Minderheit der Gene (Transkriptionsfaktoren) für die Regulation effektiv verantwortlich ist. Daher legen meine Ergebnisse nicht nur eine mögliche Erklärung für diese Eigenschaften nahe, sondern sie zeigen auch, dass eine core-periphery-artige Struktur Fitness gegenüber vielen Kriterien liefert.

Die Ensemble in diesem Kapitel sind nach nur einem einzigen Kriterium optimiert und unterliegen keinen weiteren Nebenbedingungen, als der mittleren Anzahl Eingänge je Knoten und der Kritikalität. Des Weiteren berücksichtigt mein Ansatz keine topologischen Eigenschaften, die durch Nichtgleichgewichtsprozesse entstehen, wie z. B. Genduplizierung [54, 55, 56] oder frozen accidents. Gerade daher kann diese Art der Analyse Einsichten in topologische Eigenschaften bieten, die notwendigerweise bei einer gegebenen Prozedur zu erwarten sind, und diese somit auf Konsistenz zu untersuchen. So ist etwa bekannt, dass Genduplizierung eine breite Gradverteilung [54, 55, 56] und assortative modulare Strukturen [57] zur Folge haben kann. Diese Eigenschaften sind nicht in meinen Ergebnissen enthalten, auch wenn ich sie nicht ausschließe. Dies deutet stark darauf hin, dass diese Eigenschaften nicht zwingend für robuste Systeme notwendig und ggf. nur ein Nebenprodukt von Nichtgleichgewichtswachstumsprozessen sind.

---

## 4.8 Ausblick

---

Im Laufe der Forschung zeigt sich, dass es keine triviale und eindeutige Wahl für die Funktionsentropie zu geben scheint. Vor mir wurden hier drei Möglichkeiten vorgestellt und diskutiert. Weitere Forschung könnte hier anschließen und weitere Möglichkeiten untersuchen bzw. begründen, was eine gute kanonische Wahl wäre.

Darüber hinaus konnte ich in Simulationen sehen, dass es für noch größere als die von mir untersuchten mittleren Grade  $K$  weitere Phänomene gibt. Da die Numerik hier allerdings zunehmend instabiler wurde, habe ich diese Bereiche bislang nicht weiter untersucht. Zudem wird es schwierig, mittlere Gerade  $K \gtrsim 1000$  durch Anwendungen zu motivieren. Dies gilt ebenso für die nicht ausführlich untersuchten kleinen Grade  $2 < K \lesssim 2 + 10^{-3}$ . Nichtsdestoweniger könnte auch hier weitere Forschung ansetzen. Außerdem scheint meine Numerik innerhalb der untersuchten Bereiche in der *sonstig* Phase noch Schwachstellen aufzuweisen.

Des Weiteren ließen sich die analytischen Modelle, mit denen ich Vorhersagen für die Phasenübergänge treffe, noch verfeinern. Insbesondere habe ich bislang keine gute Beschreibung für den optimalen Zustand des Ensembles, der im Falle von kleineren  $K$  und  $\mu = 1$  angenommen wird, gefunden.

Auch wäre es interessant zu wissen, inwieweit sich Blocknetzwerke mit skalenfreien Netzwerken aus Kapitel 3 kombinieren ließen. Dies würde zum einen eine komplexere Modellierung erfordern und zum anderen ggf. andere Skalenexponenten liefern. Mitunter ließen sich sogar mittels Konzepten der Blocknetzwerke eine Korrelation zwischen Ein- und Ausgangsgrad herstellen, indem je Grad ein eigener Block zur Beschreibung genutzt würde.

---

## 5 Ergebnis

In dieser Arbeit habe ich mich mit kritischen Booleschen Netzwerken befasst. Dabei ging es vor allem um Untersuchungen spezieller Netzwerkstrukturen.

Die Skaleneigenschaften der Netzwerke, vor allem die der nicht-gefrorenen Knoten, lassen sich numerisch effizient und sogar analytisch mit der Containermethode untersuchen. Hier konnte ich in Kapitel 2 aufzeigen, dass diese Methode für große mittlere Grade  $K \approx 5$  scheitert. Das heißt, dass sich das Einfrieren eines Netzwerkes nicht mehr als eine Art Lawine ausgehend von Knoten mit konstanter Funktion auffassen lässt. Trotzdem ließen sich keine qualitativen Unterschiede zwischen dem gefrorenen Kern von Netzwerken mit  $K < 5$  und  $K \geq 5$  finden. Mittels einer Mean-Field-Überlegung konnte ich das Einfrierverhalten der Netzwerke sehr gut reproduzieren und ebenfalls ein Skalengesetz für die Verteilung der Zustandswechsel auf der Transiente begründen.

In Kapitel 3 habe ich skalenfreie Netzwerke untersucht, bei denen die Ausgangsgradverteilung einem Potenzgesetz mit divergierendem zweiten Moment [25] unterlag. Dies konnte ich kombinieren mit unterschiedlichen Eingangsgradverteilungen, unter anderem auch skalenfreien. Hierfür musste ich die eingangs erwähnte Containermethode auf beliebige Ausgangsgradverteilungen erweitern. Insgesamt konnte ich eine große Anzahl an Skalengesetzen herleiten und numerisch überprüfen. Es zeigte sich, dass die Einflüsse des Potenzgesetz-Exponenten für die Ein- und Ausgangsgradverteilungen entgegengesetzte Einflüsse auf den Skalenexponenten der nicht-gefrorenen Knoten bzw. der relevanten Knoten haben. Auch hat die genaue Methode, mittels derer Stichproben aus den Verteilungen gewonnen werden, einen starken Einfluss auf die Exponenten. Des Weiteren unterscheiden sich die Eigenschaften der Attraktoren der Netzwerke mit skalenfreier Ausgangsgradverteilung deutlich von denen mit Poissonverteilung.

Es wäre interessant, noch Netzwerke mit Korrelationen zwischen Ein- und Ausgangsgrad zu untersuchen. Einige Gedanken zur Umsetzbarkeit finden sich in Abschnitt 3.9. Vielleicht ließe sich der Fragestellung auch mithilfe von geeigneten Blocknetzwerken nachgehen, siehe dazu Abschnitt 4.8.

Im dritten Teil meiner Arbeit in Kapitel 4 habe ich Boolesche Netzwerke mit dem Konzept der stochastischen Blocknetzwerke kombiniert. Nachdem ich den Begriff der Kritikalität analytisch auf Blocknetzwerke übertragen und diskutiert hatte, lag der Fokus auf dem Finden von *Nullmodellen*, also kritischen Netzwerken maximaler Entropie. Je nach Gewichtung von Entropie der Netzwerktopologie und Entropie der Aktualisierungsfunktionen haben sich qualitativ andere Lösungen ergeben. Neben Modellen für die gefundenen Phasen sind auch die Eigenschaften der Phasen als solches interessant

---

und wurden mit Netzwerktopologien aus anderen Forschungsgebieten verglichen.

In Abschnitt 4.8 habe ich interessante Anknüpfungspunkte für zukünftige Arbeiten zu Booleschen Blocknetzwerken diskutiert. So konnte etwa das Phasendiagramm bisher leider für mittleren Graden sehr nahe an 2 bzw. bei sehr großen Graden noch nicht untersucht werden, da hier die Numerik zunehmend instabiler wird. Selbst innerhalb des Phasendiagramms gibt es vereinzelt Phasenübergänge die unter Umständen nur von einer unzureichenden Numerik herrühren. Um diese untersuchen zu können, müsste man die bereits jetzt sehr aufwendige Optimierung noch weiter verbessern. Des Weiteren konnte bisher nicht abschließend geklärt werden, welche Funktionsentropie die kanonische Wahl ist. Resultate, die hier zu einer eindeutigen Festlegung kämen, ggf. sogar auf eine bisher noch nicht untersuchte Funktionsentropie, wären sehr spannend.

---

# A Anhang

---

## A.1 Spektralradius von nichtnegativen reellen Matrizen

---

Sei  $A = (a_{i,j}) \in \mathbb{R}^{n \times n}$  eine Matrix mit nichtnegativen Einträgen  $A \geq 0$ . Sei

$$\varrho(A) = \lim_{m \rightarrow \infty} \sqrt[m]{\|A^m\|} \quad (\text{A.1})$$

$$= \lim_{m \rightarrow \infty} \sqrt[m]{\|A^m\|_1} \quad (\text{A.2})$$

$$= \lim_{m \rightarrow \infty} \sqrt[m]{\max_{j=1,\dots,n} \sum_{i=1}^n |(A^m)_{i,j}|} \quad (\text{A.3})$$

der Spektralradius von  $A$ . Jeder der Terme

$$(A^m)_{i,j} = \sum_{i_1, i_2, \dots, i_{m-1}=1}^n a_{i,i_1} \left( \prod_{k=1}^{m-2} a_{i_k, i_{k+1}} \right) a_{i_{m-2}, j} \quad (\text{A.4})$$

besteht ausschließlich aus positiven Summanden. Die Summe lässt sich anschaulich als Summe über alle möglichen Pfade der Länge  $m$  von  $j$  nach  $i$  auffassen. Für  $m > n$  kommen diese Pfade an manchen Indizes zwingend mehrfach vorbei. Das Produkt der dazwischen liegenden Einträge bezeichne ich als einen geschlossenen Pfad. Sei  $z = a_{c_1, c_2} a_{c_2, c_3} \cdots a_{c_f, c_1}$  der geschlossene Pfad, bei dem das geometrische Mittel  $\sqrt[f]{z}$  maximal wird. Für  $m \rightarrow \infty$  werden die Produkte aus abzählbar unendlich vielen geschlossenen Pfaden zusammengesetzt. Für genügend große  $m$  gilt, dass jene Summanden, in denen  $z$  am häufigsten vorkommt, ihre jeweilige Summe für  $n \rightarrow \infty$  dominieren. Eine solche Summe hat daher die Form

$$(A^m)_{i,j} = z^{\frac{m}{f}} c_1 + c_2. \quad (\text{A.5})$$

Da  $c_2$  nach Definition von  $z$  höchstens genauso schnell mit  $m$  wächst wie  $z$ , kann dies im Falle von identischem Wachstum durch Anpassung von  $c_1$  zusammengefasst werden. Falls es (exponentiell) langsamer wächst, fällt es hingegen bei der Grenzwertbildung weg. Durch die Wahl von  $z$  ist auch die



Wahl des maximalen  $j$  in der Definition von  $\varrho$  trivial. Wir erhalten nun im Grenzwert

$$\varrho(A) = \lim_{m \rightarrow \infty} \sqrt[m]{z^{\frac{m}{f}} c_1} \quad (\text{A.6})$$

$$= \lim_{m \rightarrow \infty} \sqrt[m]{z^{\frac{m}{f}}} \sqrt[m]{c_1} \quad (\text{A.7})$$

$$= z^{\frac{1}{f}}. \quad (\text{A.8})$$

Bisher musste ich unendlich viele Kandidaten untersuchen, um  $z$  bestimmen zu können. Tatsächlich reicht es, alle möglichen Pfade zu untersuchen, die an jedem Knoten maximal einmal vorbeikommen (*schleifenfrei*), da sich aus dieser Menge alle anderen Pfade multiplikativ zusammensetzen lassen. Diese Menge bezeichne ich mit  $P(n)$ . Durch das Multiplizieren (Zusammensetzen) wird das geometrische Mittel nie größer als das von einem der Multiplikatoren (Komponenten). Da ich  $n$  unterschiedliche Knoten habe, ist diese Menge beschränkt. Sie enthält

$$|P(n)| = \sum_{m=1}^n \binom{n}{m} \frac{m!}{m} = \sum_{m=1}^n \frac{n!}{(n-m)!m} \quad (\text{A.9})$$

zu untersuchende Pfade. D. h. ich kann den Spektralradius von nichtnegativen Matrizen berechnen, indem ich in

$$\varrho(A) = \max_{(i_1, \dots, i_m) \in P(n)} \sqrt[m]{a_{i_1, i_2} a_{i_2, i_3} \cdots a_{i_m, i_1}} \quad (\text{A.10})$$

alle möglichen schleifenfreien geschlossenen Pfade betrachte. Derjenige mit dem größten geometrischen Mittel bestimmt den Spektralradius.

Für  $n = 2$  ergibt sich

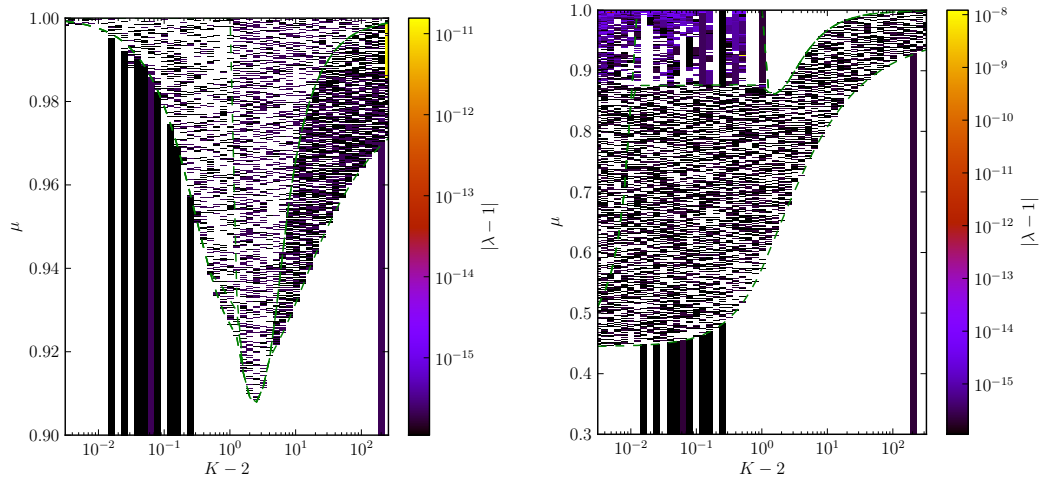
$$\varrho(A) = \max \left( a_{1,1}, a_{2,2}, \sqrt{a_{1,2} a_{2,1}} \right). \quad (\text{A.11})$$



---

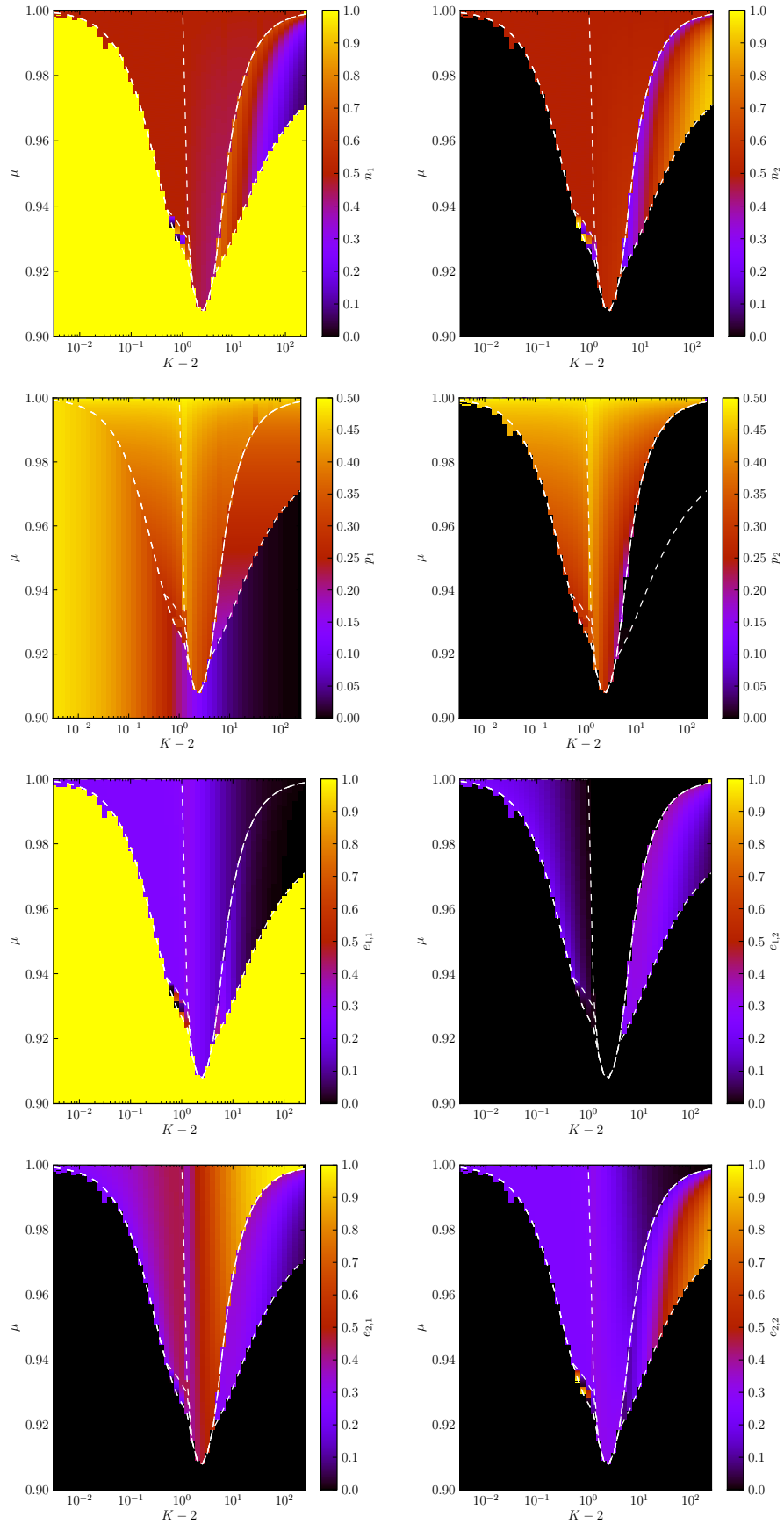
## A.2 Details zu optimalen Lösungen der kritischen Blocknetzwerke

---

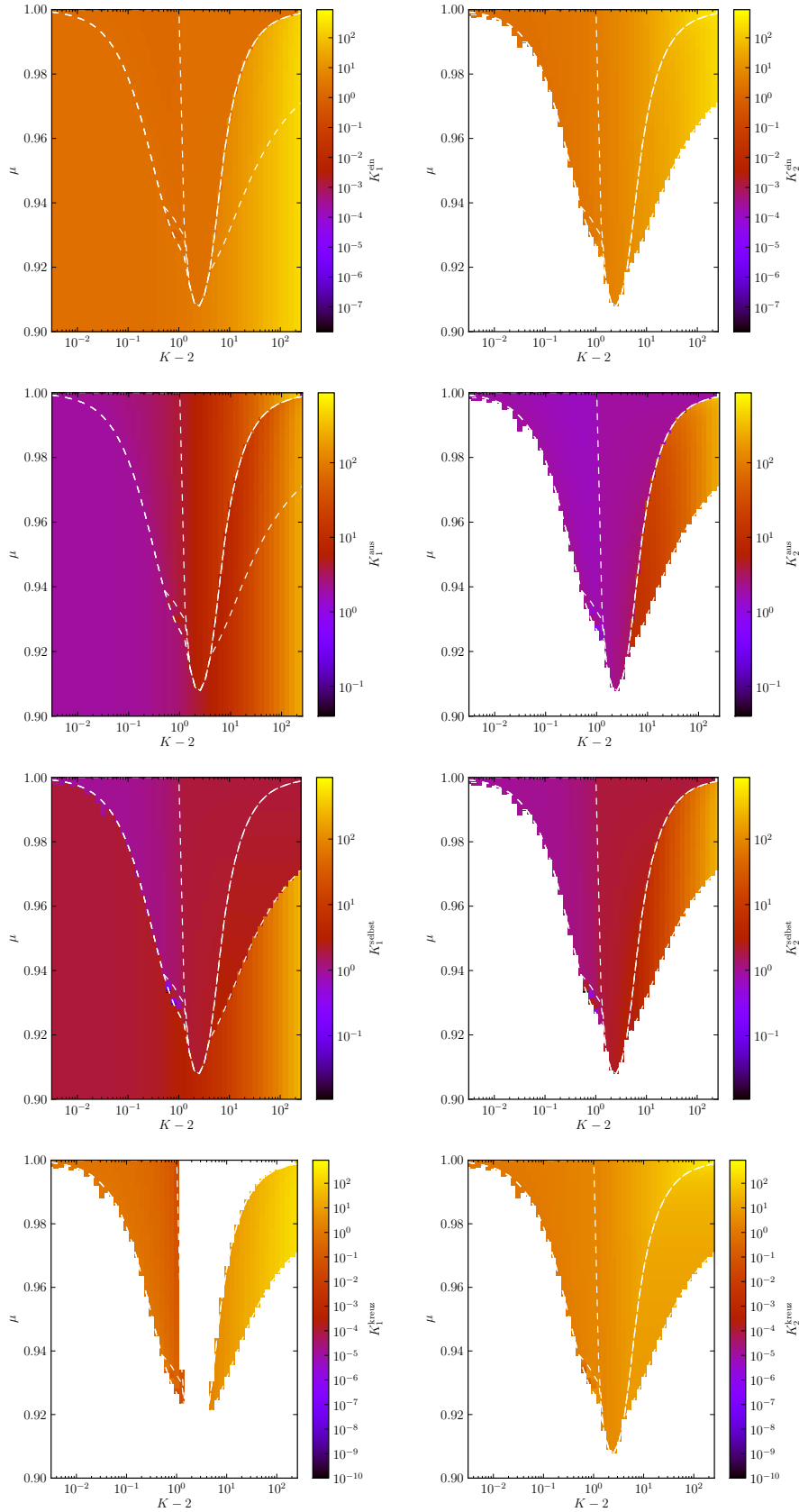


**Abbildung A.1.: Güte der Kritikalität  $|\lambda - 1|$  der optimalen Lösungen**

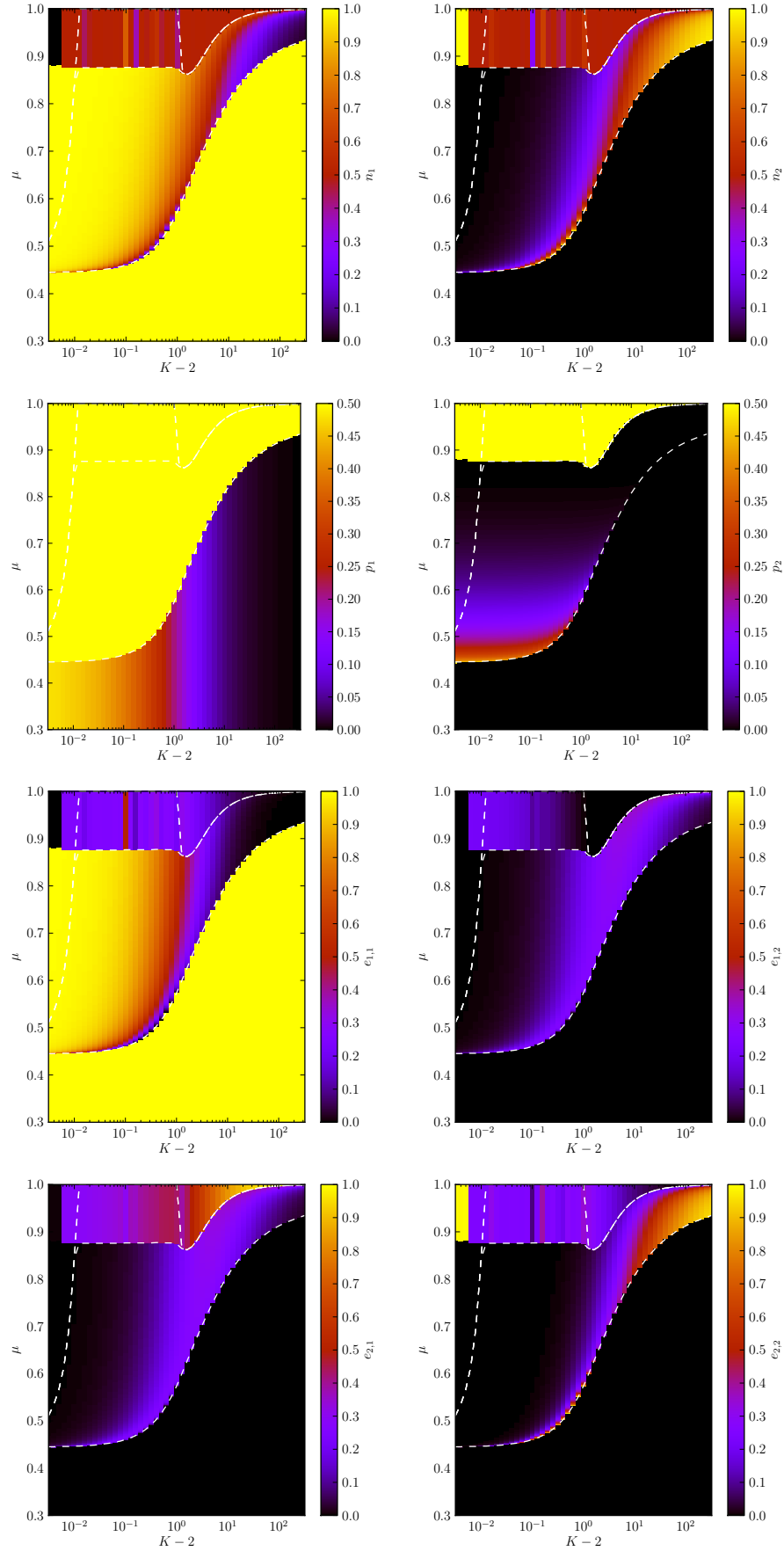
Links:  $S_F = S_F^B$ , Rechts:  $S_F = S_F^P$ . Weiße Bereiche sind im Rahmen der numerischen Genauigkeit exakt 0. Das schwarz/weiße Rauschen zeigt, dass die Güte der Kritikalität im Rahmen der möglichen numerischen Genauigkeit lag.



**Abbildung A.2.: Variablen der optimalen Lösungen für  $S_F = S_F^B$**   
 Zeilenweise von oben nach unten:  $n_1$ ,  $n_2$ ,  $p_1$ ,  $p_2$ ,  $e_{1,1}$ ,  $e_{1,2}$ ,  $e_{2,1}$ ,  $e_{2,2}$ . Man erkennt, dass manche Parameter in manchen Phasen homogen sind. Des Weiteren ist bei vielen Phasenübergängen nur ein Teil der Parameter unstetig.

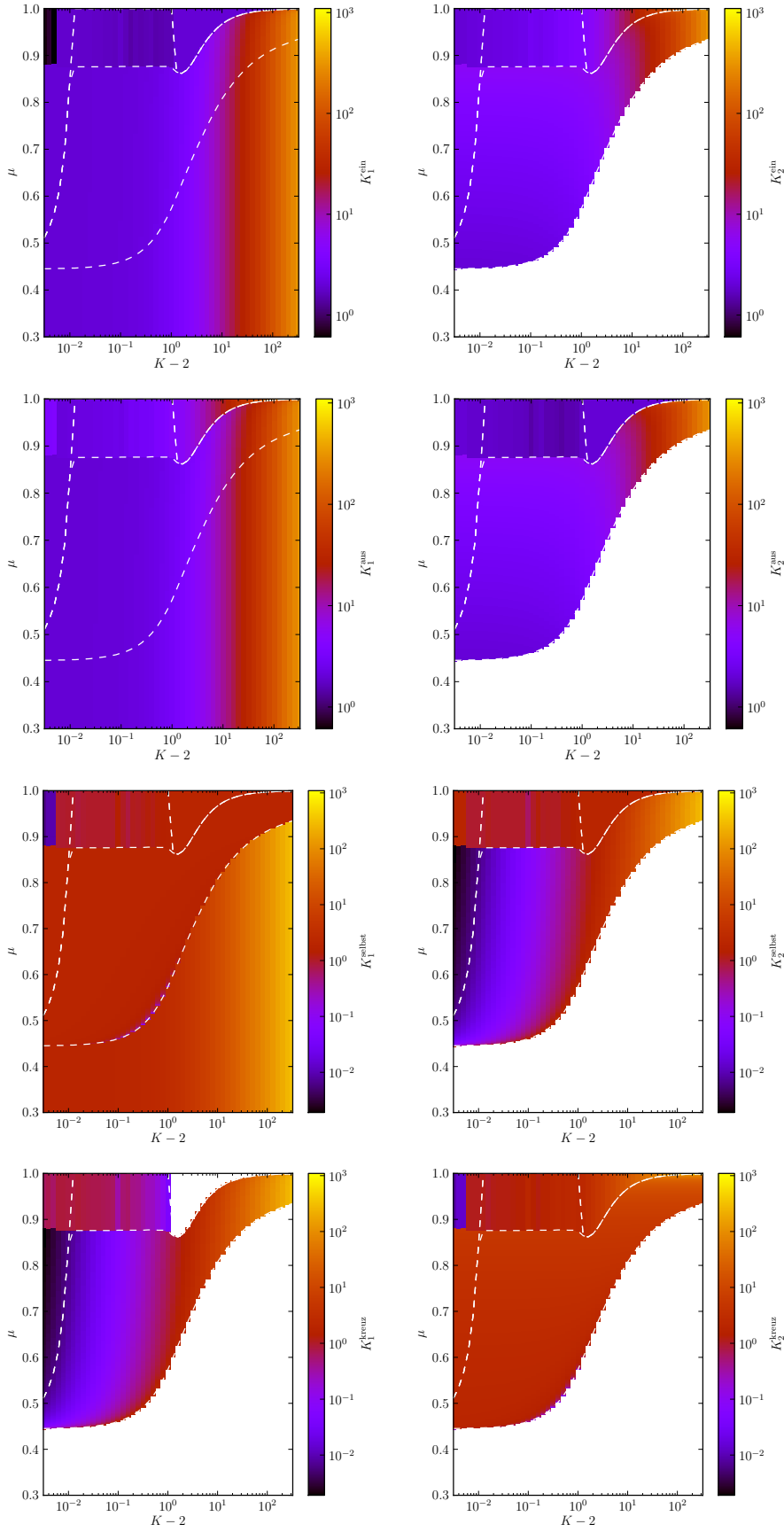


**Abbildung A.3.: Abgeleitete Größen der optimalen Lösungen für  $S_F = S_F^B$**   
 Zeilenweise von oben nach unten: links für Block  $r = 1$ , rechts für Block  $r = 2$ .  
 $K_r^{\text{ein}} = \sum_s \frac{E_{r,s}}{N_r}$ ,  $K_r^{\text{aus}} = \sum_{r'} \frac{E_{r',r}}{N_r}$ ,  $K_r^{\text{selbst}} = \frac{E_{r,r}}{N_r}$ ,  $K_r^{\text{krenz}} = \frac{E_{3-r,r}}{N_r}$ . Weiße Bereiche haben einen undefinierten Wert.



**Abbildung A.4.: Variablen der optimalen Lösungen für  $S_F = S_F^P$**

Zeilenweise von oben nach unten:  $n_1$ ,  $n_2$ ,  $p_1$ ,  $p_2$ ,  $e_{1,1}$ ,  $e_{1,2}$ ,  $e_{2,1}$ ,  $e_{2,2}$ . Man erkennt, dass manche Parameter in manchen Phasen homogen sind. Des Weiteren ist bei vielen Phasenübergängen nur ein Teil der Parameter unstetig.



**Abbildung A.5.: Abgeleitete Größen der optimalen Lösungen für  $S_F = S_F^p$**   
 Zeilenweise von oben nach unten: links für Block  $r = 1$ , rechts für Block  $r = 2$ .  
 $K_r^{\text{ein}} = \sum_s \frac{E_{r,s}}{N_r}$ ,  $K_r^{\text{aus}} = \sum_{r'} \frac{E_{r',r}}{N_r}$ ,  $K_r^{\text{selbst}} = \frac{E_{r,r}}{N_r}$ ,  $K_r^{\text{kreuz}} = \frac{E_{3-r,r}}{N_r}$ . Weiße Bereiche haben einen undefinierten Wert.

# Abbildungsverzeichnis

1.1	Beispiel für ein Boolesches Netzwerk . . . . .	4
1.2	Beispiel für ein Boolesches Netzwerk auf einem Attraktor . . . . .	8
1.3	Skizze des Containeralgorithmus . . . . .	12
2.1	Visualisierung der Wahrscheinlichkeitskomponenten aus Gleichung (2.2) . . . . .	16
2.2	Wahrscheinlichkeitsverteilung der Anzahl der nicht-gefrorenen Knoten $N_{nf}$ für $K = 4$ . . . . .	17
2.3	Größe der Container über die Zeit . . . . .	18
2.4	Anzahl der Knoten $N_0$ in Container $C_0$ . . . . .	18
2.5	Einfriedynamik der echten Dynamik verglichen mit dem Containeralgorithmus . . . . .	20
2.6	Anteil der gefrorenen Knoten als Funktion der Zeit . . . . .	20
2.7	Anteil der Knoten, die nicht für verschiedene Anfangsbedingungen auf den gleichen Wert einfrieren . . . . .	23
2.8	Anteil $p_1$ der Knoten, die auf den Wert 1 einfrieren . . . . .	23
2.9	Mittlere Dauer (x-Achse) und Varianz dieser Dauer (y-Achse), bis ein Knoten zum letzte Mal seinen Zustand ändert bei Variation des Startzustandes des Netzwerks . . . . .	25
2.10	Mittelwert und Varianz von ganzen Zahlen . . . . .	25
2.11	Wahrscheinlichkeitsverteilung für die Anzahl der Zustandswechsel von Knoten auf der Transiente . . . . .	26
2.12	Vergleich der Mean-Field-Überlegung mit simulierten Daten für die Zustandswechselwahrscheinlichkeit in Abhängigkeit von der Zeit . . . . .	29
3.1	Skizze des Containeralgorithmus . . . . .	36
3.2	Grafische Darstellung der 9 Fälle aus Tabelle 3.1 . . . . .	43
3.3	Typisches Beispiel für die Wahrscheinlichkeitsverteilung für die Anzahl der nicht-gefrorenen Knoten . . . . .	46
3.4	Legende für die Farbkodierung in Abbildung 3.5 und 3.6 anhand von Beispielen . . . . .	47
3.5	Numerische Ergebnisse zum Skalenexponenten $x$ . . . . .	48
3.6	Numerische Ergebnisse zum Skalenexponenten $y$ . . . . .	49
3.7	Klassenhierarchie der wichtigsten Klassen, siehe Abschnitt 3.7 . . . . .	50
4.1	Beispiel für ein gerichtetes Blocknetzwerk . . . . .	57
4.2	Vergleich der unterschiedlichen Funktionsentropien $S_F$ . . . . .	62
4.3	Schematischer Ablauf des Maximierungsalgorithmus für konstantes $K$ und variables $\mu$ . . . . .	67



4.4	Phasendiagramm für kritisches Blocknetzwerk . . . . .	68
4.5	Exemplarische, optimale Lösungen aus unterschiedlichen Pha- sen aus Abbildung 4.4 für $S_F = S_F^B$ . . . . .	69
4.6	Relativer Abstand der Strukturentropie $S_S$ (oben) und Funkti- onsentropie $S_F^B$ (unten) von ihrem jeweiligen, maximalen Wert für jeweiliges $K$ . . . . .	71
4.7	Qualitatives Beispiel einer Pareto-Front . . . . .	72
4.8	Konvexe Hüllen der Pareto-Fronten aller gefundenen Lösungen	73
4.9	Modelle für den kritische Punkt $\mu_c$ , des Übergangs zwischen komplett zufälliger Phase $B = 1$ und einer strukturierten Phase	75
4.10	Maximaler Unterschied in der Struktur- oder Funktionsentro- pie für festes $K$ und beliebiges $\mu \in [0, 1]$ in Abhängigkeit von $K$ . . . . .	75
A.1	Güte der Kritikalität $ \lambda - 1 $ der optimalen Lösungen . . . . .	83
A.2	Variablen der optimalen Lösungen für $S_F = S_F^B$ . . . . .	84
A.3	Abgeleitete Größen der optimalen Lösungen für $S_F = S_F^B$ . . . . .	85
A.4	Variablen der optimalen Lösungen für $S_F = S_F^P$ . . . . .	86
A.5	Abgeleitete Größen der optimalen Lösungen für $S_F = S_F^P$ . . . . .	87

---

# Tabellenverzeichnis

1.1	Liste aller $2^{2^2} = 16$ möglichen Aktualisierungsfunktionen für einen Knoten mit 2 Eingängen und ihre Klassifizierung . . . . .	5
3.1	Tabelle der Exponenten für skalenfreie Netze . . . . .	42

---

# Literaturverzeichnis

- [1] Stuart Kauffman, Carsten Peterson, Bjørn Samuelsson, and Carl Troein. Random Boolean network models and the yeast transcriptional network. *Proc. Natl. Acad. Sci. U.S.A.*, 100:14796, 2003.
- [2] Stuart A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.*, 22:437–467, 1969.
- [3] S. Bornholdt. Less is more in modeling large genetic networks. *Science*, 310(5747):449, 2005.
- [4] Stuart Kauffman. Homeostasis and differentiation in random genetic control networks. *Nature*, 224:177–178, October 11 1969.
- [5] B. Drossel. Random boolean networks. *Reviews of nonlinear dynamics and complexity*, pages 69–110, 2008.
- [6] Stefan Bornholdt and Thimo Rohlf. Topological evolution of dynamical networks: Global criticality from local dynamics. *Physical Review Letters*, 84(26):6114–6117, June 2000.
- [7] Stefan Bornholdt and Torsten Röhl. Self-organized critical neural networks. *Physical Rev. E*, 67(6):066118, June 2003.
- [8] Matthias Rybarsch and Stefan Bornholdt. Avalanches in self-organized critical neural networks: A minimal model for the neural SOC universality class. *PLoS ONE*, 9(4):e93090, April 2014.
- [9] Tiago P. Peixoto and Barbara Drossel. Noise in random boolean networks. *Phys. Rev. E*, 79:036108, Mar 2009.
- [10] Eva Gehrmann, Christine Gläßer, Yaochu Jin, Bernhard Sendhoff, Barbara Drossel, and Kay Hamacher. Robustness of glycolysis in yeast to internal and external noise. *Phys. Rev. E*, 84:021913, Aug 2011.
- [11] Barbara Drossel, Tamara Mihaljev, and Florian Greil. Number and length of attractors in a critical Kauffman model with connectivity one. *Phys. Rev. Lett.*, 94:088701, 2005.
- [12] V. Kaufman, T. Mihaljev, and B. Drossel. Scaling in critical random Boolean networks. *Phys. Rev. E*, 72(4):046124, October 2005.
- [13] Henrik Flyvbjerg. An order parameter for networks of automata. *J. Phys. A*, 21:L955, 1988.
- [14] Henrik Flyvbjerg and N. J. Kjør. Exact solution of Kauffman’s model with connectivity one. *J. Phys. A*, 21:1695, 1988.

- 
- [15] Ugo Bastolla and Giorgio Parisi. Relevant elements, magnetization and dynamical properties in Kauffman networks. a numerical study. *Physica D*, 115(3&4):203, May 1998.
- [16] Ugo Bastolla and Giorgio Parisi. The modular structure of Kauffman networks. *Physica D*, 115(3&4):219, May 1998.
- [17] Joshua E. S. Socolar and Stuart A. Kauffman. Scaling in ordered and critical random Boolean networks. *Phys. Rev. Lett.*, 90:068702, 2003.
- [18] Tamara Mihaljev and Barbara Drossel. Scaling in a general class of critical random boolean networks. *Phys. Rev. E*, 74(4):046101, Oct 2006.
- [19] B. Derrida and Y. Pomeau. Random networks of automata: A simple annealed approximation. *Europhys. Lett*, 1(2):45–49, 1986.
- [20] Maximo Aldana-Gonzalez, Susan Coppersmith, and Leo P. Kadanoff. Boolean dynamics with random couplings. In Ehud Kaplan, Jerrold E. Mardsen, and Katepalli R. Sreenivasan, editors, *Perspectives and Problems in Nonlinear Science, A celebratory volume in honor of Lawrence Sirovich*, pages 23–89. Springer Applied Mathematical Sciences Series, Springer Verlag, New York, May 2003.
- [21] Bernard Derrida and Dietrich Stauffer. Phase transitions in two dimensional Kauffman cellular automata. *Europhys. Lett.*, 2(10):739–745, November 1986.
- [22] Stuart A Kauffman. Requirements for evolvability in complex systems: orderly dynamics and frozen components. *Physica D: Nonlinear Phenomena*, 42(1):135–152, 1990.
- [23] B. Samuelsson and C. Troein. Superpolynomial Growth in the Number of Attractors in Kauffman Networks. *Physical Review Letters*, 90(9):098701, March 2003.
- [24] B. Drossel and F. Greil. Critical boolean networks with scale-free in-degree distribution. *Physical Rev. E*, 80(2):026102, 2009.
- [25] D.S. Lee and H. Rieger. Broad edge of chaos in strongly heterogeneous boolean networks. *Journal of Physics A: Mathematical and Theoretical*, 41:415001, 2008.
- [26] Marco Möller and Barbara Drossel. The formation of the frozen core in critical boolean networks. *New Journal of Physics*, 14(2):023051, 2012.
- [27] Marco Möller and Barbara Drossel. Scaling laws in critical random boolean networks with general in-and out-degree distributions. *Physical Rev. E*, 87(5):052106, 2013.
- [28] R. Albert. Scale-free networks in cell biology. *Journal of cell science*, 118(21):4947–4957, 2005.

- 
- [29] Jeffrey J. Fox and Colin C. Hill. From topology to dynamics in biochemical networks. *Chaos*, 11:809–815, December 2001.
- [30] Shu-ichi Kinoshita, Kazumoto Iguchi, Hiroaki Yamada, Yamada Tokuyama, Michio Tokuyama, Irwin Oppenheim, and Hideya Nishiyama. Robustness of attractor states in complex networks. *AIP Conference Proceedings*, 982:768–771, February 2008.
- [31] Maximino Aldana. Boolean dynamics of networks with scale-free topology. *Physica D*, 185:45, 2003.
- [32] P. Oikonomou and P. Cluzel. Effects of topology on network evolution. *Nature Physics*, 2(8):532–536, 2006.
- [33] M. Aldana and P. Cluzel. A natural class of robust networks. *Proceedings of the National Academy of Science*, 100:8710–8714, July 2003.
- [34] Piotr Fronczak, Agata Fronczak, and Janusz A. Holyst. Kauffman boolean model in undirected scale-free networks. *Phys. Rev. E*, 77:036119–5, March 2008.
- [35] K. Iguchi, S. Kinoshita, and H.S. Yamada. Boolean dynamics of kauffman models with a scale-free network. *arXiv preprint cond-mat/0510430*, 2005.
- [36] R. Serra, M. Villani, and L. Agostini. On the dynamics of random Boolean networks with scale-free outgoing connections. *Physica A Statistical Mechanics and its Applications*, 339:665–673, August 2004.
- [37] W. Kahan. Pracniques: Further remarks on reducing truncation errors. *Commun. ACM*, 8(1):40–, January 1965.
- [38] V. Kaufman and B. Drossel. Relevant components in critical random boolean networks. *New Journal of Physics*, 8(10):228, 2006.
- [39] Marco Möller and Tiago P Peixoto. Maximum-entropy large-scale structures of boolean networks optimized for criticality. *New Journal of Physics*, 17(4):043021, 2015.
- [40] Eva Ackermann. *Vergleich von Boolescher und kontinuierlicher Dynamik auf Genregulationsnetzwerken*. PhD thesis, TU Darmstadt, Jan 2013.
- [41] Eva Ackermann, Eva Marie Weiel, Torsten Pfaff, and Barbara Drossel. Boolean versus continuous dynamics in modules with two feedback loops. *The European Physical Journal E*, 35(10):107, 1–13, 2012.
- [42] Eva Gehrmann and Barbara Drossel. Boolean versus continuous dynamics on simple two-gene modules. *Physical Review E*, 82(4):046120, 1–9, 2010.
- [43] Tiago P. Peixoto. Emergence of robustness against noise: A structural phase transition in evolved models of gene regulatory networks. *Physical Rev. E*, 85(4):041908, April 2012.

- 
- [44] Eva Ackermann, Tiago P Peixoto, and Barbara Drossel. Reliable dynamics in boolean and continuous networks. *New Journal of Physics*, 14:123029, 1–17, Dezember 2012.
- [45] Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, June 1983.
- [46] Tiago P. Peixoto and Stefan Bornholdt. Evolution of robust network topologies: Emergence of central backbones. *Physical Review Letters*, 109(11):118703, September 2012.
- [47] Tiago P. Peixoto. Entropy of stochastic blockmodel ensembles. *Physical Rev. E*, 85(5):056122, May 2012.
- [48] Rodrigo Poblanno-Balp and Carlos Gershenson. Modular random boolean networks. *Artificial life*, 17(4):331–351, 2011.
- [49] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.
- [50] Stephen E. Harris, Bruce K. Sawhill, Andrew Wuensche, and Stuart Kauffman. A model of transcriptional regulatory networks based on biases in the observed regulation rules. *Complexity*, 7(4):23–40, 2002.
- [51] M. J. D. Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. *Advances in optimization and numerical analysis*, 7:51–67, 1994.
- [52] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, January 1965.
- [53] Nicola Beume, Boris Naujoks, and Michael Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, September 2007.
- [54] Romualdo Pastor-Satorras, Eric Smith, and Ricard V. Solé. Evolving protein interaction networks through gene duplication. *Journal of Theoretical Biology*, 222(2):199–210, May 2003.
- [55] I. Ispolatov, P. L. Krapivsky, and A. Yuryev. Duplication-divergence model of protein interaction network. *Physical Rev. E*, 71(6):061911, June 2005.
- [56] Fan Chung, Linyuan Lu, T. Gregory Dewey, and David J. Galas. Duplication models for biological networks. *Journal of Computational Biology*, 10(5):677–687, October 2003.
- [57] Jakob Enemark and Kim Sneppen. Gene duplication models for directed networks with limits on growth. *Journal of Statistical Mechanics: Theory and Experiment*, 11:7, November 2007.

---

# Danksagung

An erster Stelle bedanke ich mich bei meiner Betreuerin Barbara Drossel für die Bereitstellung des Forschungsthemas und dafür, dass ich jederzeit mit Fragen oder Ideen zu ihr kommen konnte. Insbesondere bedanke ich mich für die Freiheiten und Flexibilität, die sie mir ermöglicht hat, um meine Forschung parallel zu Familie und Firmengründung zu betreiben. Ebenfalls bedanke ich mich bei meinem Betreuer für den Blocknetzwerkteil dieser Arbeit, Tiago Peixoto, für die Bereitstellung des Themas sowie die Zusammenarbeit bei der Strukturierung und Diskussion der Ergebnisse. Ich danke Burkhard Dünweg für die Begutachtung dieser Arbeit.

Auch danke ich meiner gesamten Arbeitsgruppe für die vielen Aufmunterungen und Ablenkungen zwischendurch und die schöne gemeinsame Zeit, insbesondere Yixian Song und Korinna Allhoff, mit denen ich immer gerne ein Büro geteilt habe.

Meinen Kollegen bei der Administration des Computer Clusters der Arbeitsgruppe danke ich für die gute Teamarbeit und die gegenseitige Unterstützung. Dies waren über die Zeit hinweg: Daniel Löb, Tiago Peixoto, Sebastian Plitzko, Christian Kolb, Christioph Schmitt und Philipp Gramlich.

Da ich parallel zur Promotion an der Gründung und dem Aufbau meiner Firma MAVinci gearbeitet habe, danke ich auch meinen Mitgründern Johanna Claussen, Cornelius Claussen und Michael Niesen für den Freiraum, die sie mir und meiner Promotion eingeräumt haben.

Für das Korrekturlesen dieser Arbeit bedanke ich mich bei Katharina Brand, Frederik Brand, Denis Möller, Dankrad Feist, Wolf Behrenhoff, Peter Schauß und meiner Frau Claudia.

Insbesondere danke ich meiner Familie, also meiner Frau Claudia und meinen Töchtern Eva-Lotte, Stine und Lisabeth für die Rücksichtnahme und Unterstützung speziell in den arbeitsintensiven Phasen dieser Promotion.

Dank geht auch an die DFG, die meine Arbeit an skalenfreien Netzen unter der Zuwendungsnummer Dr300/4-2 unterstützt hat.

---

# Lebenslauf

---

## Persönliche Daten

---

Name	Marco Möller
Wohnort	Evenaristraße 67, 64293 Darmstadt
Geburtsdatum und -ort	08. Juni 1982, Hildesheim
Staatsangehörigkeit	deutsch / brasilianisch
Familienstand	verheiratet, 3 Kinder (1, 3, 6 Jahre)

---

## Bildungsweg

---

1988 - 1992	Grundschule Borsum
1992 - 1994	Orientierungsstufe Harsum
1994 - 1998	Albertus-Magnus-Realschule (Hildesheim): erweiterter Sekundarabschluss I. Note: 1,8
1998 - 2001	Bosch/Blaupunkt: Ausbildung zum Kommunikationselektroniker Fachrichtung Funktechnik. Note: 1,0
1999 - 2003	6 Jugend forscht-Teilnahmen mit insgesamt 3 verschiedenen Themen: Planung und Simulation eines selbstorganisierenden Funknetzwerkes: Höchste Platzierung 2. Platz Bundeslandebene im Bereich Mathematik / Informatik Semiautonomes Fernerkundungssystem: Höchste Platzierung 2. Platz Bundeslandebene im Bereich Technik Dynamisches Fahrgastinformationssystem: Höchste Platzierung 3. Platz Deutschlandebene im Bereich Technik
2001 - 2002	Fachoberschule Technik (Werner-von-Siemens-Schule Hildesheim): Fachhochschulreife. Note: 1,4
Mai 2003	Praktikum im Institut für Theoretischen Physik an der TU Braunschweig, AG Prof. Wolfram Brenig: Perkolation auf 2,5-dimensionalen Gittern
2003 - 2004	Studium der Informatik an der Universität Kassel: Erwerb der Hochschulzugangsberechtigung. Note: 1,2
2004 - 2008	Doppelstudium Physik BSc. (Note: 1,2) und Informatik BSc. (Note: 1,3) an der TU Darmstadt Abschlussarbeiten geschrieben bei Dr. Daniel Polani an der University of Hertfordshire (UK) mit den Titeln:



---

	Informatik: Information Theoretical Models for Hierarchical Structured Agents and their Computation
	Physik: Information Maximizing Agents in a Structured Environment
2008	1. Platz beim Ideenwettbewerb UniTechSpin-Award der TU Darmstadt mit MAVinci, anschließend Unternehmensgründung
2008 - 2010	Studium Physik MSc. an der TU Darmstadt. Note: 1,1 Abschlussarbeit: Proof of Concept of Topography Measurement with a Micro Airborne Lidar System Aboard a Micro Air Vehicle
seit 2010	Promotion in der theoretischen Physik bei Prof. Dr. Barbara Drossel an der TU Darmstadt über: Eigenschaften von kritischen Booleschen Zufallsnetzwerken für insgesamt 14 Monaten unterbrochen zwecks Elternzeiten

---

# Publikationsliste

---

## Begutachtete Veröffentlichungen

---

- Möller, M. & Polani, D. 2008. Common concepts in agent groups, symmetries, and conformity in a simple environment. In S. Bullock, J. Noble, R. Watson, and M. A. Bedau, editors, *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pages 420–427. MIT Press, Cambridge, MA, 2008.
- Polani, D. & Möller, M. 2009. Models of information processing in the sensorimotor loop. In *Information Theory and Statistical Learning*, pages 289–308. Springer, 2009.
- Möller, M. & Drossel, B. 2012. The formation of the frozen core in critical Boolean networks. *New Journal of Physics*, 14(2):023051, 2012.
- Möller, M. & Drossel, B. 2013. Scaling laws in critical random Boolean networks with general in-and out-degree distributions. *Physical Review E*, 87(5):052106, 2013.
- Möller, M. & Peixoto, T. 2015. Maximum-entropy large-scale structures of boolean networks optimized for criticality. *New Journal of Physics*, 17(4):043021, 2015.

---

## Tagungsbeiträge

---

- Möller, M. & Polani, D. 2008. Common Concepts in Agent Groups, Symmetries, and Conformity in a Simple Environment. Vortrag auf der Artificial Life XI Konferenz 2008 in Winchester UK.
- Möller, M. & Drossel, B. 2012. Formation of the Frozen Core in Critical Boolean Networks. Vortrag auf der Frühjahrstagung der Deutschen Physikalischen Gesellschaft 2012 in Berlin.
- Möller, M. 2012. Unmannend Aerial Systems. Forum CeBIT in Motion 2012 in Hannover.
- Möller, M. 2012. Practical Showcase: MAVinci SIRIUS Orthophoto UAS competes traditional surveying methods. Intergeo TMF 2012 in Hannover.

- 
- Möller, M. 2013. UAS in der Praxis: Von der Projektplanung bis zur virtuellen Vermessung. UAS in der Praxis – 18. Münchner Fortbildungsseminar Geoinformationssysteme 2013 in München.
  - Möller, M. & Drossel, B. 2013. Scaling Laws in Critical Random Boolean Networks with General in- and out-degree Distributions. Vortrag auf der Frühjahrstagung der Deutschen Physikalischen Gesellschaft 2013 in Regensburg.
  - Möller, M. 2013. UAS in der Praxis: Von der Projektplanung bis zur virtuellen Vermessung. VDV Seminar Leistungspotentiale von UAV's 2013 in Fulda.
  - Möller, M. 2013. Surveying UAS and Accuracy: Professional Flight Planning and More. Vortrag auf dem Intergeo Kongress sowie dem Intergeo Trend- und Medienforum 2013 in Essen.